

**Design of two digital radiation tolerant  
Integrated Circuits for High Energy  
Physics experiments data readout.**

Titolo italiano:

**Progetto di due circuiti integrati digitali  
resistenti a radiazione per la lettura di  
dati da esperimenti di fisica delle alte  
energie.**

Relatore:  
Prof. Ing. Giovanni Verzellesi

Tesi di:  
Sandro Bonacini

Correlatori:  
Dr. Alessandro Marchioro  
Dr. Kostas Kloukinas

Controrelatore:  
Prof. Ing. Fausto Fantini

Anno Accademico 2001/2002



*A mio padre*



# Contents

<b>Abstract</b>	<b>ix</b>
<b>Riassunto in lingua italiana</b>	<b>1</b>
I. Dall'esperimento CMS al preshower . . . . .	2
II. Dispositivi a semiconduttore e circuiti integrati resistenti a radiazione . . . . .	2
III. Il Kchip . . . . .	4
IV. Una RAM statica in tecnologia 0.13 micron resistente a radiazione	5
<b>1 Introduction</b>	<b>7</b>
1.1 CERN and High Energy Physics . . . . .	7
1.1.1 Accelerators and detectors . . . . .	7
1.1.2 The Large Hadron Collider . . . . .	8
1.2 The Compact Muon Solenoid experiment . . . . .	9
1.2.1 The Trigger and Data Acquisition System . . . . .	12
1.2.2 The Electromagnetic Calorimeter . . . . .	13
1.2.3 The preshower . . . . .	14
1.3 Notation . . . . .	15
<b>2 Radiation tolerant semiconductor devices and integrated circuits</b>	<b>17</b>
2.1 Introduction . . . . .	17
2.1.1 Radiation environment in the LHC . . . . .	17
2.1.2 Radiation environment in the CMS experiment . . . . .	18
2.1.3 Radiation tolerant ICs . . . . .	20
2.2 Radiation effects . . . . .	21
2.2.1 Radiation effects on matter . . . . .	21
2.2.2 Radiation effects on electrical parameters of MOS transistors . . . . .	22
2.2.3 Single Event Effects (SEE) . . . . .	26
2.3 Hardening against radiation . . . . .	27
2.3.1 Layout techniques . . . . .	28
2.3.2 Circuit and system techniques . . . . .	29

2.4	A radiation tolerant digital standard cells library . . . . .	33
2.4.1	The CMOS 0.25 $\mu$ m library . . . . .	33
<b>3</b>	<b>The Kchip</b>	<b>35</b>
3.1	The CMS preshower front-end system . . . . .	35
3.1.1	The control logic . . . . .	36
3.1.2	Fast timing control signals . . . . .	38
3.1.3	The silicon detector . . . . .	39
3.1.4	The PACE chipset . . . . .	39
3.1.5	The AD41240 ADC . . . . .	42
3.1.6	The Gigabit Optical Link chip . . . . .	42
3.1.7	The I <sup>2</sup> C interface . . . . .	45
3.2	The Kchip . . . . .	46
3.2.1	Functionalities . . . . .	46
3.2.2	Design tools and techniques . . . . .	47
3.2.3	Design flow . . . . .	49
3.2.4	Top-level block diagram . . . . .	50
3.2.5	Buffers size . . . . .	50
3.2.6	The Clock and Control block and synchronization goals	54
3.2.7	The Trigger Decoder . . . . .	56
3.2.8	The Trigger Handler . . . . .	58
3.2.9	The PACE Controller . . . . .	61
3.2.10	The Error Logger . . . . .	65
3.2.11	The DeDDR . . . . .	66
3.2.12	The Data FIFO . . . . .	66
3.2.13	The Column Addresses FIFO . . . . .	67
3.2.14	The Packet Formatter . . . . .	68
3.2.15	The GOL Interface . . . . .	73
3.2.16	The CalPulse Builder . . . . .	76
3.2.17	The I2C Block . . . . .	77
3.2.18	Testability . . . . .	80
3.2.19	Synthesis . . . . .	81
3.2.20	Input/output pads . . . . .	82
3.2.21	Floorplanning . . . . .	84
3.2.22	Place & Route . . . . .	86
3.3	The Kchip prototype . . . . .	86
<b>4</b>	<b>A radiation tolerant CMOS 0.13 micron Static RAM</b>	<b>89</b>
4.1	Architecture . . . . .	89
4.2	The memory cell . . . . .	90
4.2.1	Sizing the cell . . . . .	91
4.2.2	SPICE simulations . . . . .	93
4.2.3	Cell layout . . . . .	94
4.3	The SRAM core . . . . .	95

4.3.1	The Word-line Decoder . . . . .	97
4.3.2	The Bit-line driver . . . . .	97
4.3.3	The Read Logic . . . . .	99
4.4	Self-timing technique . . . . .	99
4.5	Input/output blocks . . . . .	103
4.6	Final layout . . . . .	104
4.7	Future development and improvement . . . . .	107
<b>A</b>	<b>The Delay Locked Loop</b>	<b>109</b>
<b>B</b>	<b>A triplicated parameterized CRC generator</b>	<b>111</b>
	<b>Bibliography</b>	<b>117</b>





# Abstract

High Energy Physics research (HEP) involves the design of readout electronics for its experiments, which generate a high radiation field in the detectors. The several integrated circuits placed in the future Large Hadron Collider (LHC) experiments' environment have to resist the radiation and carry out their normal operation.

In this thesis I will describe in detail what, during my 10-months participation in the digital section of the Microelectronics group at CERN, I had the possibility to work on:

- The design of a radiation-tolerant data readout digital integrated circuit in a  $0.25\ \mu\text{m}$  CMOS technology, called “the Kchip”, for the CMS preshower front-end system. This will be described in Chapter 3.
- The design of a radiation-tolerant SRAM integrated circuit in a  $0.13\ \mu\text{m}$  CMOS technology, for technology radiation testing purposes and future applications in the HEP field. The SRAM will be described in Chapter 4.

All the work has carried out under the supervision and with the help of Dr. Kostas Kloukinas and the section leader Dr. Alessandro Marchioro.



# Riassunto in lingua italiana

Il lavoro per questa tesi è stato svolto in circa 10 mesi presso il gruppo di microelettronica del CERN, l'Organizzazione Europea per la Ricerca Nucleare situata a Ginevra (Svizzera). Un nuovo acceleratore di particelle è tuttora in fase di costruzione al CERN: si tratta del Large Hadron Collider (LHC), il cui completamento è previsto per l'anno 2007. Questo acceleratore sarà in grado di creare collisioni protone-protone altamente energetiche fino a 14 TeV, un traguardo mai raggiunto finora, che permetterà di rispondere a domande fondamentali sulla fisica.

Quattro grandi esperimenti saranno costruiti per sfruttare questa risorsa: CMS, ATLAS, LHCb ed ALICE. Essi sorgeranno in quattro distinti punti lungo l'acceleratore, ove le collisioni di particelle avranno luogo. Ciascun esperimento sarà composto da più rivelatori con differenti caratteristiche e finalità, ma ciascun rivelatore conterrà, almeno in parte, elettronica necessaria per la lettura e trasmissione dei dati al di fuori dell'esperimento. Poiché l'ambiente in cui questi circuiti saranno inseriti diverrà, a seguito delle innumerevoli collisioni di particelle, altamente radioattivo, si rende necessario concepire circuiti elettronici capaci di sopportare tali radiazioni per tutta la durata prevista di funzionamento di LHC, il che vale a dire 10 anni.

Vista l'altissima quantità di informazione da estrarre da ogni rivelatore, l'utilizzo di circuiti integrati dedicati risulta indispensabile. Questi dispositivi devono essere sviluppati con accorgimenti particolari per rispondere ai suddetti vincoli. Il lavoro svolto per questa tesi è il progetto di due circuiti integrati digitali resistenti a radiazione, rispettivamente in due differenti tecnologie CMOS:

- un dispositivo per lettura dei dati dal preshower, uno dei rivelatori dell'esperimento CMS, realizzato in una tecnologia  $0.25 \mu\text{m}$  e chiamato "Kchip";
- una RAM statica, realizzata in una tecnologia  $0.13 \mu\text{m}$ , per misure di resistenza a radiazione ed eventuale successivo impiego come macrocella in altri circuiti integrati da inserire in ambiente altamente radioattivo.

## I. Dall'esperimento CMS al preshower

L'esperimento chiamato Compact Muon Solenoid (CMS) [7], è formato essenzialmente da una parte interna, più vicina al punto di collisione delle particelle, con compiti di tracciamento delle traiettorie, da una parte intermedia con compiti calorimetrici, in altre parole di misurazione dell'energia, ed infine da una parte esterna di rivelazione dei muoni. Il calorimetro più interno è un calorimetro elettromagnetico (ECAL) [8] basato su cristalli di scintillazione in tungstenato di piombo ( $\text{PbWO}_4$ ), in grado di trasferire l'energia di elettroni incidenti sotto forma di luce rilevabile da fotodiodi e fototriodi posti sulla superficie esterna. Un sottile rivelatore, detto preshower, adiacente ai cristalli, ma più interno rispetto ad essi, si fa carico di misurare con buona accuratezza la posizione di entrata dell'elettrone convertito per ottenere una misura della sua energia.

Il preshower è composto da rivelatori a strisce in silicio, contenenti ciascuno 32 diffusioni  $p^+$  su substrato n che corrono lungo tutta la lunghezza del dispositivo. Le giunzioni sono poste in polarizzazione inversa in modo che una particella carica entrante nella regione di svuotamento che abbia sufficiente energia ionizzi gli atomi del reticolo, scalzando elettroni dalla banda di valenza e creando lacune. Tali portatori saranno catturati dal campo elettrico e costituiranno quindi una corrente nel dispositivo. Un banco di preamplificatori posti nelle immediate vicinanze del rivelatore si fa carico di misurare la carica raccolta da ciascuna striscia. I dati così raccolti devono, attraverso vari stadi, infine raggiungere un veloce collegamento in fibra ottica uscente dall'esperimento per arrivare ad un potente sistema di calcolo e raccolta di informazioni chiamato *counting room*. Il sistema, posto nelle vicinanze del rivelatore e dentro l'esperimento, che si fa carico di misurare e inviare i dati al link ottico è chiamato *sistema di front-end*.

## II. Dispositivi a semiconduttore e circuiti integrati resistenti a radiazione

Gli effetti dovuti a radiazione nei semiconduttori si possono riassumere in due classi principali [5]: ionizzazione e spostamento. Il primo effetto consiste nella creazione di coppie elettrone-lacuna, ed il numero di coppie generate è direttamente proporzionale alla dose totale assorbita<sup>1</sup>. Lo spostamento è invece un danno diretto al reticolo del cristallo in cui uno degli atomi viene mosso dalla sua sede, spesso per finire in una posizione interstiziale (difetto di Frenkel). I transistor MOS sono poco sensibili ai difetti di spostamento,

---

<sup>1</sup>La dose è una grandezza fisica che misura l'assorbimento di energia per unità di superficie irradiata. L'unità di misura S.I. è il *Gray*, ove  $1 \text{ Gy} = 1 \text{ J/Kg}$ . Ciò nonostante, nell'ambiente della fisica delle alte energie, la vecchia unità *rad* è ancora usata:  $1 \text{ rad} = 10^{-2} \text{ Gy} = 100 \text{ erg/g}$ .

## II. Dispositivi a semiconduttore e circuiti integrati resistenti a radiazione 3

poichè la conduzione avviene su un sottile strato in prossimità della superficie del semiconduttore e non in profondità, dove la maggior parte dei danni si addensa<sup>2</sup>.

L'ossido di silicio presente in ogni transistor MOS ed usato anche come isolante fra dispositivi nelle tecnologie integrate è tuttavia molto sensibile agli effetti di ionizzazione [1]: mentre gli elettroni generati dalla radiazione incidente hanno una mobilità sufficiente ad essere velocemente espulsi dall'ossido, le lacune rimangono in gran parte intrappolate. In particolare, nel caso di un transistor a canale n con gate polarizzato positivamente, le lacune raggiungono lentamente l'interfaccia ossido-semiconduttore e lì si fermano, abbassando la tensione di soglia del dispositivo. Allo stesso tempo le radiazioni inducono la formazione di trappole all'interfaccia, le quali invece tendono ad alzare la tensione di soglia. Il bilancio diventa positivo per tecnologie moderne con ossido di gate sottile. Per un transistor a canale p, eventuali lacune intrappolate nell'ossido abbassano la tensione di soglia, che viene però alzata in modulo, e l'effetto delle trappole all'interfaccia è dello stesso segno. In conclusione, in seguito a irradiazione la tensione di soglia si alza in modulo per tutti i dispositivi MOS.

Tuttavia, nelle tecnologie commerciali moderne come la  $0.25\ \mu\text{m}$ , in cui l'ossido di gate è molto sottile, le variazioni della tensione di soglia indotte da radiazione sono contenute ed accettabili fino ed oltre a 30 Mrad di dose totale assorbita. Il vero problema risulta invece essere causato dalla degradazione dell'ossido di campo e dell'ossido laterale<sup>3</sup>, i quali, essendo spessi accumulano quantità maggiori di carica, pregiudicando in certe zone la loro capacità di isolamento elettrico. Infatti, quando cresciuto al di sopra di un substrato p, l'ossido contenente carica positiva può invertire la popolazione del semiconduttore sottostante creando un percorso conduttivo. Ciò pregiudica il funzionamento dei transistori a canale n, aumentando con la dose totale assorbita la corrente di perdita tra drain e source.

Accorgimenti particolari di layout e di circuito possono risolvere queste problematiche senza dovere ricorrere a tecnologie specializzate con processi dedicati, che ai fini pratici sono meno avanzate ed hanno un costo elevato ed una resa inferiore alle tecnologie commerciali.

Questi accorgimenti risultano in pratica nel modificare la forma di tutti i transistori MOS a canale n, eliminando il bordo del canale e così l'ossido laterale, ed in più aggiungere ad essi un anello di guardia  $p^+$  circostante per isolarli da altri dispositivi. Il gate, che diviene di forma circolare, circonda quindi completamente il drain, posto al suo interno, mentre il source rimane

---

<sup>2</sup>Altri dispositivi, come ad esempio i transistori bipolari, sono più sensibili a questo tipo di difetti proprio perché basati su una conduzione verticale.

<sup>3</sup>L'*ossido laterale* risiede ai bordi del canale e sotto il gate. Nelle tecnologie moderne la tecnica Shallow Trench Isolation (STI) è utilizzata al posto della più anziana LOCOS. L'accezione *ossido di campo* è qui utilizzata riferendosi all'ossido di isolamento tra dispositivi differenti, indipendentemente dalla tecnica utilizzata.

all'esterno (si veda la fig. 2.7 a pag. 28). Tutto questo si traduce ovviamente in una riduzione della massima densità di componenti per unità di area.

Una libreria di standard cells [19] per il veloce sviluppo automatico di parti digitali di circuiti integrati è stata preparata dal gruppo di microelettronica del CERN, rispettando le considerazioni precedenti.

Un'ulteriore difficoltà nel disegno di circuiti digitali per ambiente radioattivo è la vulnerabilità ai Single Event Upset (SEU) da parte della logica: una particella carica può interagire con un dispositivo in modo da cambiare temporaneamente il livello logico della sua uscita. Ne segue che il contenuto di qualsiasi dispositivo integrato di memorizzazione può subire cambiamenti e va quindi protetto con un'opportuna ridondanza a livello di circuito. Ciò è maggiormente importante per i registri utilizzati nelle macchine a stati finiti, utilizzate sovente nelle logiche di controllo dei circuiti integrati: un dato errato in uno di tali registri può indurre comportamenti errati da parte dell'intero circuito, e quindi del sistema. La soluzione utilizzata è di triplicare ciascuna macchina a stati finiti ed introdurre una logica di voto sui tre dati in uscita da esse: il dato scelto da almeno 2 macchine su 3 verrà posto in uscita. Lo stesso ragionamento è applicato per lo stato memorizzato, che viene votato e ricaricato in ognuna delle tre macchine a stati finiti con accorgimenti particolari (si veda la sezione 2.3.2 a pag. 29).

### III. Il Kchip

Il sistema di front-end del preshower [18] è costituito da una parte analogica connessa con il rivelatore a microstrisce di silicio, una parte di conversione analogico/digitale ed una parte digitale dedicata alla trasmissione dei dati ed al controllo del sistema. Il sistema di acquisizione lavora campionando i dati ad una frequenza di 40.08 MHz, che corrisponde esattamente al numero di scontri tra particelle per unità di tempo. I dati provenienti da un particolare scontro sono detti *evento*.

I dati, nel loro percorso dalla sorgente (il rivelatore) alla destinazione (il link ottico), attraversano nell'ordine:

- un chipset analogico, chiamato PACE, composto da un amplificatore charge-sensitive ed una memoria analogica organizzata a pipeline;
- un convertitore analogico/digitale (ADC);
- il Kchip, incaricato di memorizzare e formattare i dati in un pacchetto opportuno per la trasmissione;
- il Gigabit Optical Link (GOL) chip, che serializza i dati e pilota il laser per il link.

Ci sono quindi due stadi di memorizzazione consecutivi nel sistema di front-end: uno in forma analogica residente nel PACE chipset ed uno in forma

digitale nel Kchip. Il primo stadio si rende necessario poichè non tutti gli eventi acquisiti devono essere trasferiti al di fuori dell'esperimento: la quantità di informazione che sarebbe infatti necessario trasmettere richiederebbe un'infrastruttura ancora più imponente e costosa. Ciò che avviene è invece una selezione degli eventi, basata su un insieme limitato di dati e semplici algoritmi, scartando i dati meno interessanti e riducendo il numero di eventi inviati per unità di tempo a 100 kHz. Chiaramente la decisione sulla selezione o meno di un evento richiede un certo intervallo di tempo per essere valutata, e durante questo intervallo i dati devono essere memorizzati.

Il Kchip unisce i dati provenienti da un massimo di 4 rivelatori in un unico pacchetto, al fine di massimizzare l'utilizzo del canale. In questo modo però, il collegamento ottico risulta essere più lento dell'insieme di 4 sorgenti in termini di massime prestazioni. Ne segue la necessità del secondo stadio di memorizzazione, che pone i dati digitalizzati provenienti da 4 PACE chipsets in code FIFO, in attesa di essere inviati nel canale.

Oltre ai menzionati principali incarichi, il Kchip svolge svariate altre funzioni: controlla i 4 chipset PACE ad esso connessi e ne verifica la sincronizzazione ed il corretto funzionamento, interagisce con il chip GOL effettuando un controllo ad alto livello del link, segnala eventuali errori alla counting room, ecc. (si veda la sezione 3.2 a pag. 46 per una lista completa delle funzionalità).

L'integrato è stato progettato utilizzando tecniche CAD di sintesi e layout automatiche, facendo uso della libreria di porte logiche standard cells in tecnologia CMOS 0.25  $\mu\text{m}$  disponibile. Tutto il circuito è stato perciò descritto in Verilog, un linguaggio di descrizione dell'hardware (HDL), per poi essere passato ai tool di sintesi e place & route. Ad ogni stadio di progettazione, verifiche intensive tramite simulazione sono state effettuate grazie alle potenzialità offerte dal Verilog. Alla fine, gli usuali controlli di rispetto delle regole di layout (DRC) e corrispondenza tra layout e schematic (LVS) sono stati compiuti.

Il risultato è un chip di dimensioni  $6 \times 5 \text{ mm}^2$ , contenente 13000 porte logiche più circa 80 kbit di memoria statica (SRAM), che formano insieme un totale di circa 660000 transistori. Il chip sta per iniziare la prima fase di fabbricazione a scopo di test.

## IV. Una RAM statica in tecnologia 0.13 micron resistente a radiazione

La tecnologia 0.13  $\mu\text{m}$  risulta relativamente nuova per le applicazioni in ambiente altamente radioattivo, ed è ancora in fase di sperimentazione. Numerose misure dovranno essere compiute per qualificarla come adatta, nonostante le promesse siano molto buone: infatti l'ossido di gate è ancora più sottile della già assestata tecnologia 0.25  $\mu\text{m}$ , e gli isolamenti sono di miglio-

re qualità. Tra le misure necessarie compare la sensibilità ai SEUs, destinata ad aumentare rispetto a tecnologie precedenti: diminuendo le dimensioni dei dispositivi, si abbassano conseguentemente anche le capacità associate, e quindi una particella carica interagente con esse indurrà variazioni di tensione più elevate.

Per effettuare questi test è richiesta, una memoria statica resistente agli altri effetti di radiazione. La SRAM così ottenuta potrà poi essere utilizzata come macrocella in altri circuiti integrati per ambiente radioattivo, entrando a far parte di una libreria di celle resistenti a radiazione già parzialmente preparata al CERN.

Il progetto è basato su un disegno esistente nella precedente tecnologia [20], mantenendo la medesima architettura, ma chiaramente cambiando i dimensionamenti ed i layout. In particolare il layout della SRAM è fatto in gran parte manualmente usando solo per alcuni blocchi le standard cells, che comunque sono piazzate e connesse ancora una volta manualmente.

L'architettura scelta prevede l'utilizzo di una cella di memoria a 6 transistori single-ported in modo da risparmiare area, ma allo stesso tempo di avere un comportamento dual-ported visto dall'esterno. Ciò implica il portare a termine le due operazioni di lettura e scrittura una dopo l'altra ma nel medesimo ciclo di clock.

In più, visto il possibile riutilizzo del blocco inserito in altri integrati, la dimensione della SRAM deve essere facilmente riconfigurabile: ciò è ottenuto costruendo una memoria modulare in modo che replicando blocchi di base tante volte quanto necessario si raggiunge la grandezza di memoria voluta. La temporizzazione della memoria deve adattarsi di conseguenza alla nuova dimensione, per cui tecniche di self-timing sono state sfruttate.

Tra gli strumenti utilizzati per il progetto risalta il grande impiego di simulazioni SPICE, sia per il dimensionamento che per la verifica. Il risultato ottenuto è un blocco di dimensioni  $553 \times 129 \mu\text{m}^2$  posto in un chip di  $1.84 \times 1.96 \text{ mm}^2$ , dimensione imposta dal numero di pad input/output necessari. Si stima che la SRAM possa lavorare senza problemi fino ad una frequenza di 140 MHz (nel caso peggiore). Il chip è attualmente in fase finale di fabbricazione in un numero limitato di pezzi, grazie allo spazio concesso in un Multi-Project Wafer (MPW).



# Chapter 1

## Introduction

### 1.1 CERN and High Energy Physics

High Energy Physics (HEP) explores the innermost basic constituents of matter and their mutual interactions.

CERN<sup>1</sup>, the European Laboratory for Particle Physics, was founded in 1954 in Geneva (Switzerland) as a joint European effort to provide a major scientific facility for particle physicists. It is today one of the world's largest and most successful scientific laboratories, as well as an outstanding example of international collaboration between its 19 member states<sup>2</sup>.

#### 1.1.1 Accelerators and detectors

Particle physics studies are based on particle collisions at high kinetic energy, which means that the particles used in the experiments should have high speed. Particle accelerators, like a synchrotron, are used to reach the speed needed.

Inside particle accelerators, beams of charged particles are pushed by high frequency electrodes into a vacuum pipe. The pipe can be linear or circular: in the second case, beam bending is performed by dipole magnets accordingly to the Lorentz force law. Quadrupole magnets are used to focus the beam.

The results of a collision have then to be observed through a detector. A detector is usually composed by many sub-detectors with different capabilities and goals, and all of them are connected to a computer system for analysis and event reconstruction. The goal is to identify, count and trace, as many particles coming out from the collision point as possible.

---

<sup>1</sup>Once called Conseil Européen pour la Recherche Nucléaire, now is officially named as Organisation Européen pour la Recherche Nucléaire.

<sup>2</sup>Member states are Austria, Belgium, Bulgaria, Czech Republic, Denmark, Finland, France, Germany, Greece, Hungary, Italy, The Netherlands, Norway, Poland, Portugal, Slovak Republic, Spain, Sweden, Switzerland and the United Kingdom.

A detector together with its infrastructure is called *experiment*.

### 1.1.2 The Large Hadron Collider

In year 2000 the biggest CERN's accelerator, the Large Electron Positron collider (LEP), was dismantled in order to leave place for a new, more powerful, machine: the Large Hadron Collider (LHC). While the LEP was able to reach electron-positron collisions with a centre of mass energy of 200 GeV, the LHC is designed to collide protons, going further up to 14 TeV.

The challenge, in modern particle physics research, is to probe at higher and higher collision energies, either because the basic constituents to be studied are only present at those energies, or because they are normally tied in complex aggregates and need those energies to split apart. Reaching high energy densities means also recreating the earliest universe conditions during the big bang. Thus, the higher collision energy we manage to create, the smaller dimension we study, and the earlier back in time we can observe.

The LEP was built in a 100 m underground tunnel, with the earth shielding its radiation, following a 27 km long ring. Such a big circumference was necessary because of the energy loss by bremsstrahlung: electrons and positrons emit photons when accelerated, therefore the same happens bending their trajectory; the less the trajectory is bent, the less energy they lose.

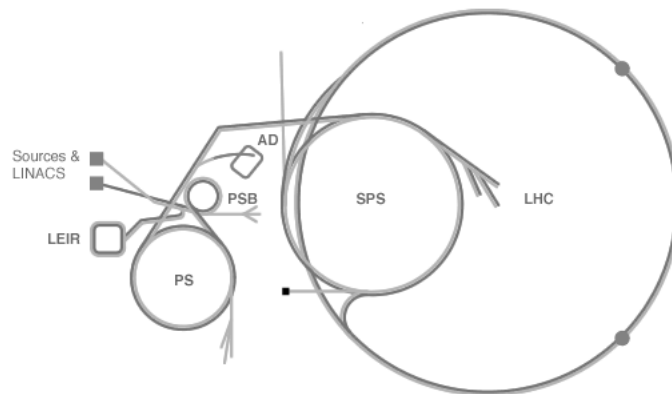


Figure 1.1: Plan of accelerators at CERN.

In these days the LEP is being replaced with the LHC, which employs the same existing cavern. The LHC is planned to be fully operational from year 2007 onward.

LHC will make use of superconducting magnets cooled at 1.9 K, spread all along the ring to bend the beams, with nominal field of 8.33 T, allowing the storage at the desired energy of 7 TeV of two proton beams. The two beams will run in opposite directions, but colliding only in four points, where

the experiments take place. LHC is designed to be able to accelerate also lead ions, much more massive than protons, to attain collision energies of 1148 TeV, but this will happen only later in the accelerator planned schedule.

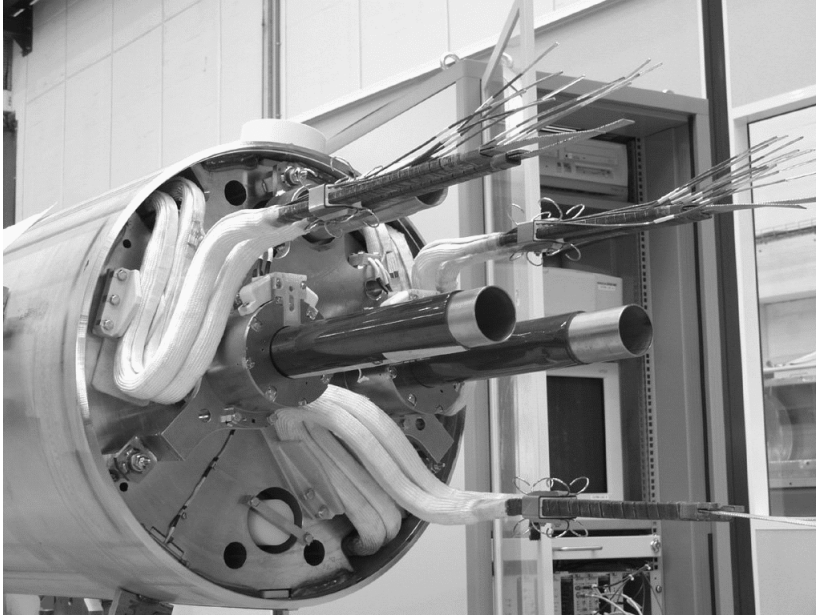


Figure 1.2: LHC accelerator section photograph.

The beams will also be segmented in 2835 bunches of  $1.1 \cdot 10^{11}$  particles per bunch. This will make two bunches running in opposite directions to meet in the interaction points every 24.95 ns at the nominal speed. In other words the collision frequency will be 40.08 MHz.

The four experiments designed to make use of LHC are:

- the Compact Muon Solenoid (CMS);
- A Toroidal Lhc ApparatuS (ATLAS);
- A Large Ion Collider Experiment (ALICE);
- LHCb.

Only the CMS experiment will be treated in detail, since most of this work has been developed to be part of it.

## 1.2 The Compact Muon Solenoid experiment

Figures 1.4 and 1.5 show representations of the CMS. As can be seen, it has a cylinder's shape with a diameter of 14.6 m and 21.6 m long, excluding the

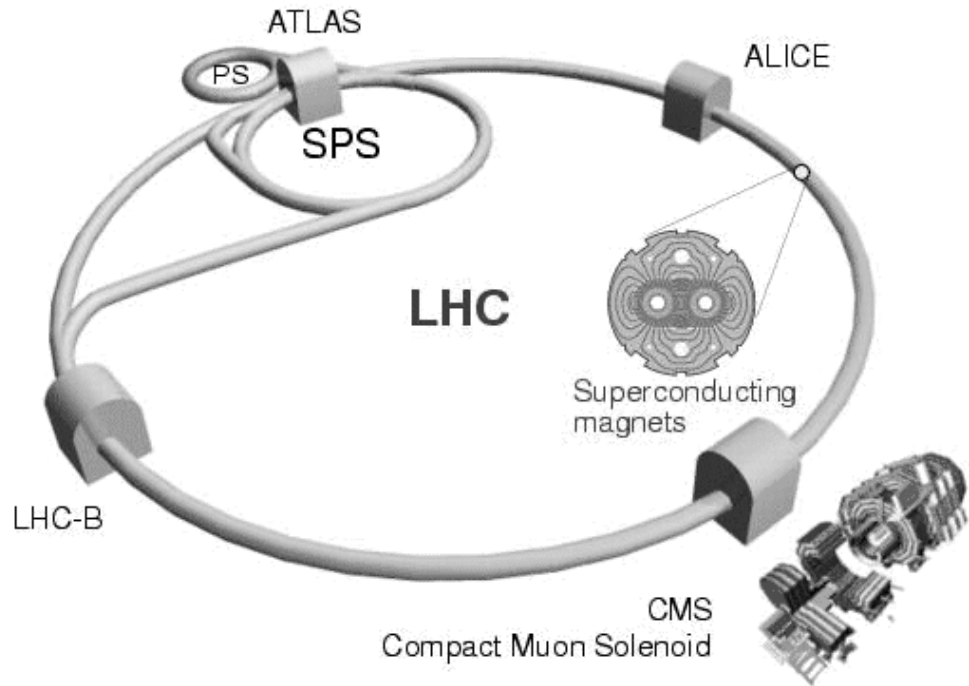


Figure 1.3: Underground view of LHC and its experiments.

very forward calorimeter. Its total weight is about 14500 tonnes. The beams run along the axis entering from the two sides, and collide in the center of the detector, the point also referred as *vertex*. The physics performance is guaranteed by its almost  $4\pi$  solid angle coverage. CMS is optimized for the Higgs boson discovery.

The detector is divided into three main sections: the middle barrel and the two side identical endcaps. A 13 m long superconducting solenoid magnet generates a uniform 4 T field inside the barrel region, which bends the charged particles' trajectory in order to identify them by their mass and charge. A return path for the magnetic flux is guaranteed by a huge iron structure, covering the whole machine, called "the return yoke". Inside the return yoke the magnetic field is of about 2 T.

The CMS is composed by several sub-detectors, which, from the inside to the outside are:

**The tracker** composed by silicon pixel detectors in the inner part, and silicon strip detectors in the outer. It traces the trajectory of charged particles with an accuracy of about  $100 \mu\text{m}$ ;

**The electromagnetic calorimeter (ECAL)** which measures the energies of electrons and photons through  $\text{PbWO}_4$  crystals. The ECAL con-

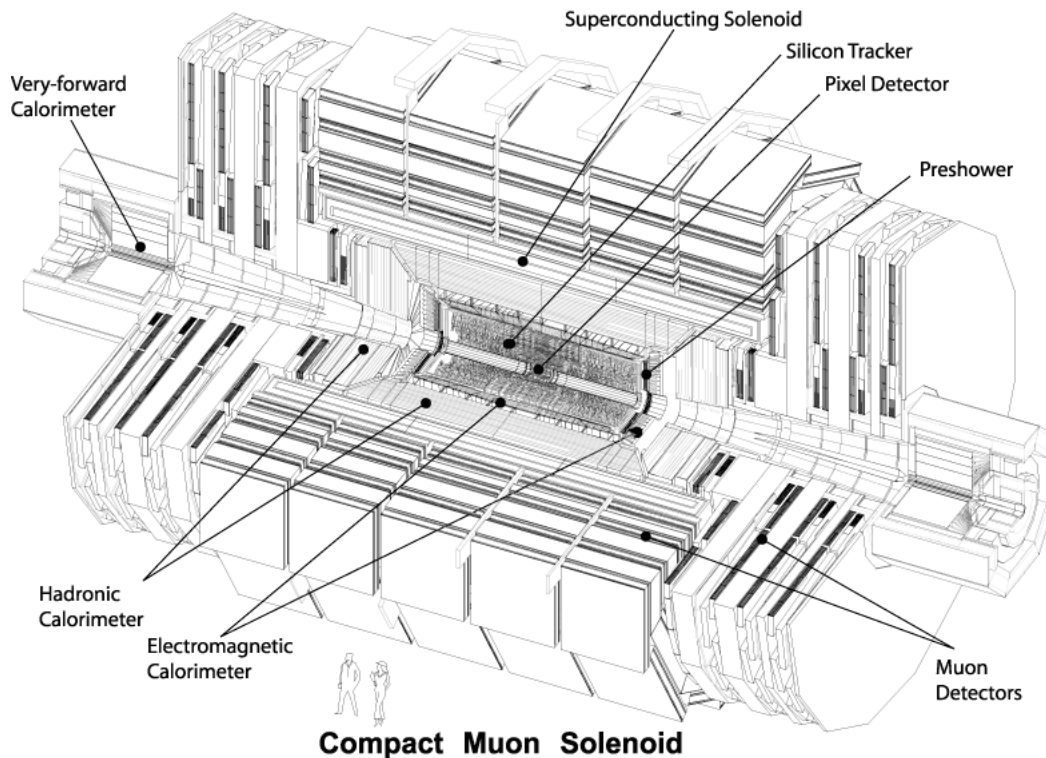


Figure 1.4: View of CMS with its parts and sub-detectors.

tains also a small silicon strip detector situated in the endcaps' inner part called the *preshower*;

**The hadronic calorimeter (HCAL)** made with thick layers of copper as absorber and thin layers of plastic scintillator, it measures the energies of hadrons;

**The muon chambers** used for detecting muons, which are highly penetrating. The muon chambers are interleaved with the iron return yoke and are made with gaseous particle detectors.

**The very forward calorimeter** placed along the axis only in the outer barrel region, it is made with an iron/gas detector.

Most of the work presented here will be part of the ECAL sub-detector preshower, thus it will be treated with more detail later.

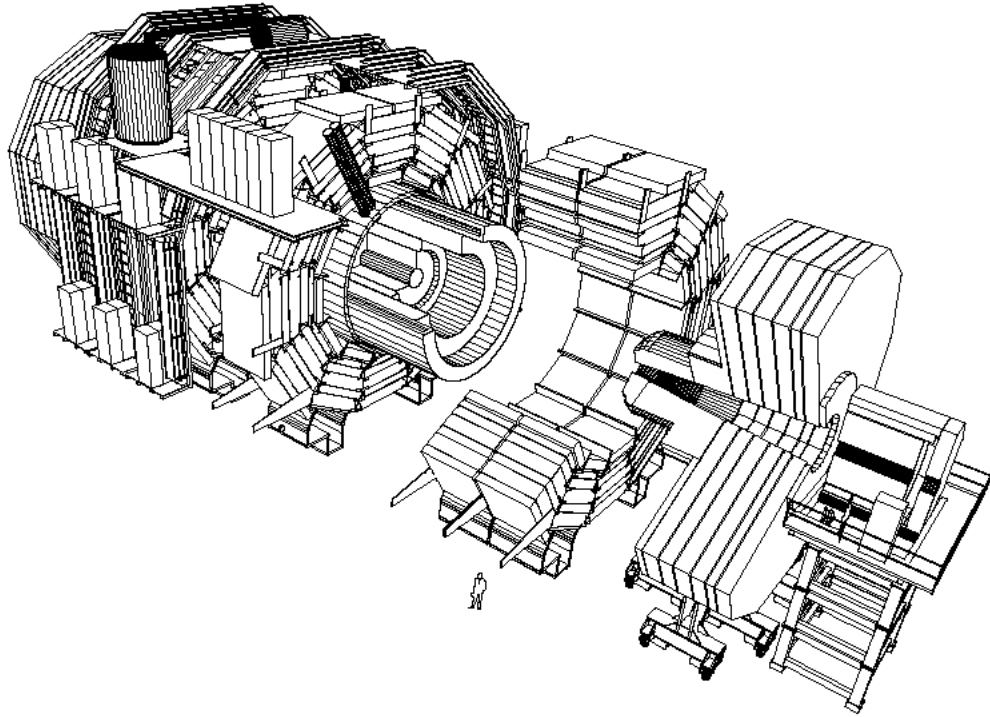


Figure 1.5: 3d split view of the CMS detector.

### 1.2.1 The Trigger and Data Acquisition System

As mentioned before, the bunch crossing frequency is about 40.08 MHz, with an average of 20 inelastic events<sup>3</sup> occurring each time. This means that a rate of about 800 MHz interactions will produce an enormous amount of data coming out from the experiment. Nevertheless, only a small fraction of the collisions will be interesting from the physics point of view, therefore a filtering of the data has to be performed. It is also necessary to do this in real-time, reducing the rate to 100 Hz, which is the maximum rate that should be archived for off-line analysis [7].

All these jobs are carried out by the Trigger and Data Acquisition System (TriDAS) of the experiment, which selects the useful events<sup>4</sup>, rejecting the rest, by evaluating a subset of the data. This operation is done in two steps, or, in other words, two subsequent selections will take place. Figure 1.6 shows the DAS block diagram.

For the first selection, the TriDAS calculates a Level-1 trigger signal indicating if the data has to be kept or not. The Level-1 trigger decision

---

<sup>3</sup>An inelastic event is collision in which the interaction involves other forces than the electromagnetic and gravitational forces.

<sup>4</sup>All the data relative to one bunch crossing is referred as an *event*.

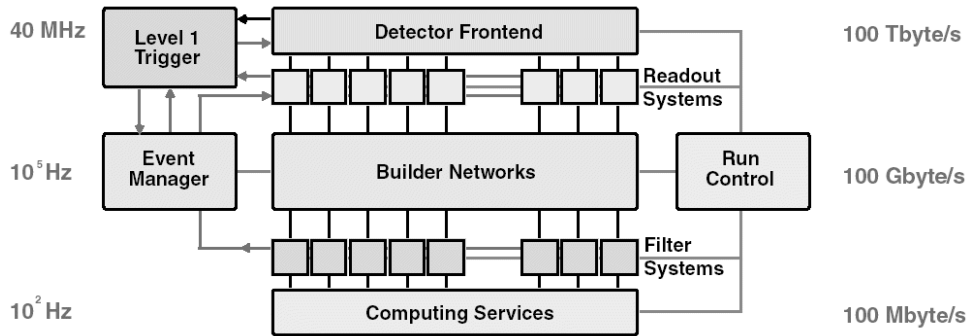


Figure 1.6: CMS Data Acquisition System block diagram.

latency is about  $3.2 \mu\text{s}$ , therefore the data coming out of the detectors has to be stored in a set of memories, waiting for it. The Level-1 trigger maximum rate is 100 kHz, thus the rejection ratio is about 400.

After the first selection, the events are reconstructed, joining together the data from different sub-detectors but relative to the same collision. Then, a Level-2 selection will be performed, obtaining the desired event rate of 100 Hz. Despite this large reduction the final experiment data rate is still over 1 Tbyte/day, which is huge for an experiment planned to last 10 years.

### 1.2.2 The Electromagnetic Calorimeter

As described before, the ECAL provides a precise measurement of the energy of photons and electrons. It has a small spatial resolution but it's designed to have an excellent energy resolution. A scintillating crystal calorimeter offer the best performance for this parameter, and, moreover, high density crystals allow a very compact calorimeter system [8].

In scintillating crystals, decelerating electrons will emit Bremsstrahlung. The generated photons will again interact with the lattice, ionizing the atoms and creating new energetic electrons and so on. This chain interaction is called *electron shower* and the resulting photons can be observed using photodetectors.

Following successful beam irradiation tests in 1994,  $\text{PbWO}_4$  (lead tungstate) crystals were chosen as the scintillating medium. The ECAL will contain 80,000 of them arranged into the barrel and endcap sections. Figure 1.7 shows a submodule unit.

The outer side of each crystal will be covered with photodetectors in order to observe the scintillation light. Silicon avalanche photodiodes (APDs<sup>5</sup>)

<sup>5</sup>APDs are diodes with a reverse biased buried pn junction at a very high electric field of sufficient strength such that photoelectrons arriving in the junction are accelerated and multiplied, by impact ionization, in an avalanche process.

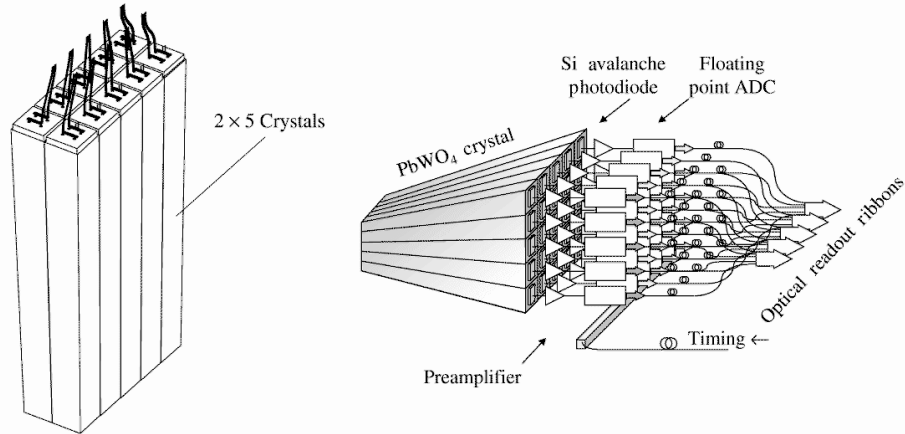


Figure 1.7: Lead tungstate crystal submodule and associated readout electronics.

have been chosen for use in the barrel region, after proving to be suitable to the high radiation and magnetic environment. Nevertheless, in the endcap regions, where the radiation environment is more harsh, vacuum phototriodes (VPTs) are used instead. The photodetectors signals are sampled by a set of ADCs and sent to the readout electronics.

### 1.2.3 The preshower

The ECAL will contain a preshower detector in the endcap regions, whose main function is to provide  $\gamma$ - $\pi^0$  separation. In fact an overlap between two showers, due to a neutral pion  $\pi^0$  decay into two photons, can be confused with a single shower from an energetic gamma ray  $\gamma$ .

The preshower measures the impact position of the shower using two orthogonal silicon strips detectors with a 1.9 mm pitch. The operating temperature of the preshower is  $-5$  °C to maintain the performance of the silicon strips after irradiation [22]. The total area covered by silicon detectors in the preshower is large:  $16.4$  m<sup>2</sup>; thus a simple classical arrangement of p<sup>+</sup> strips on n bulk structure has been chosen (the same choice has been done for the CMS tracker detector).

Each  $63 \times 63$  mm<sup>2</sup> silicon detector, subdivided into 32 strips 61 mm long, is mounted on a *micromodule* together with its front end analog electronics. A various number of micromodules form a *ladder*, which has two columns of adjacent detectors with their digital electronics mounted on top. A ladder is



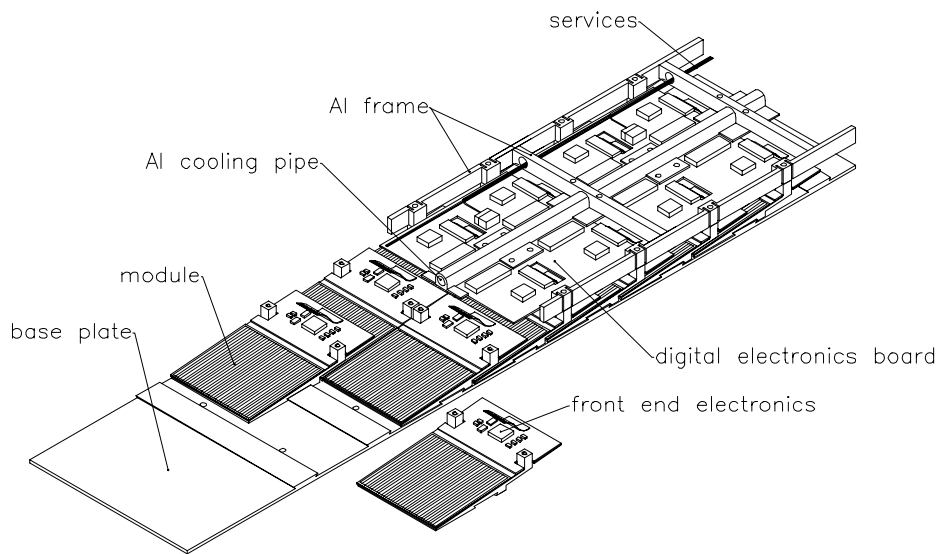


Figure 1.8: A ladder with its micromodule units.

shown in Figure 1.8. The digital electronics boards are called *motherboards* and each one of them is connected to 4 micromodules.

The data is sent out of the experiment through an 800 Mbit/s optical link. A more detailed description of the preshower's front end electronics will be given later.

### 1.3 Notation

In the following chapters a special notation will be used:

- Verbatim text will be used to indicate electrical or optical **signals** or **buses**;
- Sans serif characters will be used when referring **Functional Blocks** or **Circuits**;
- Small capital letters will be used to name STATE MACHINE'S STATES.



## Chapter 2

# Radiation tolerant semiconductor devices and integrated circuits

### 2.1 Introduction

#### 2.1.1 Radiation environment in the LHC

In order to maximize the number of interesting events obtained from the experiments, the LHC accelerator is designed to reach a very high peak luminosity<sup>1</sup>:  $10^{34} \text{ cm}^{-2}\text{s}^{-1}$  for protons and  $1.95 \cdot 10^{27} \text{ cm}^{-2}\text{s}^{-1}$  for lead ions. This will bring, in the case of protons, to an average production of  $8 \cdot 10^8$  inelastic proton-proton collisions per second, creating an extremely hostile radiation environment.

In addition, at LHC the high beam energy combined with the very high luminosity results in numerous intense cascades, which will end up in an immense number of low-energy particles. In fact, particles energies exceeding 10 GeV are expected to be very rare in the detectors' barrel, and also in the major part of the endcap. Therefore the radiation studies focussed on the energy range around 1 GeV and below.

*Studied spectra*

While induced radioactivity was negligible in electron-positron colliders (like the LEP), it will be a major concern at LHC. It can be assumed that each inelastic hadronic interaction results in a residual nucleus, which can have almost any mass and charge smaller than that of the target. Roughly 30% of the inelastic hadronic interactions create long-lived radionuclides

*Induced radioactivity*

---

<sup>1</sup>The luminosity can be thought as the number of particles per unit area in the interaction point of the two beams.

[14, 28] which contribute to the dose rate from induced activity in the experimental area. This activity decreases relatively slowly after the end of irradiation, so that even long cooling times do not significantly improve the situation. Activation can also occur through neutron interactions, especially in the thermal regime. However, except for a few special materials, this is usually a minor contribution.

### 2.1.2 Radiation environment in the CMS experiment

As resumed in Table 2.1 total dose<sup>2</sup> values in the CMS experiment could be high, in the worst case conditions, up to 50 Mrad. The detectors' front-end electronics has then to stand this enormous amount of radiation, especially in the inner tracker and in the ECAL endcaps, where the levels are higher.

<i>Sub-detector</i>		<i>Total dose</i> [Mrad]	<i>Neutron fluence</i> [ $10^{14}$ cm <sup>-2</sup> ]	<i>Charged hadron fluence</i> [ $10^{14}$ cm <sup>-2</sup> ]
Tracker	at 7cm	35.	1.	10.
	at 22cm	6.5	.35	1.5
	at 75cm	.7	.15	.25
ECAL	barrel	.5	.5	.005
	endcaps	20.	10.	.6
HCAL	barrel	.02	.1	-
	endcaps	2.5	5.	-
Muon chambers		.005	.025	-
Forward calorimeter		500.	250.	-
Experimental hall		.0005	.001	-

Table 2.1: CMS sub-detectors' radiation environment in the 10 years experiment lifetime [12], equivalent to  $5 \cdot 10^7$  s. The reported doses and fluences are the maxima inside each sub-detector.

#### *Electromagnetic calorimeter*

The silicon layer of the preshower detector will be exposed to the neutron albedo from the electromagnetic calorimeter, while charged hadrons do not contribute significantly at any preshower position. Dose rates drop rapidly when moving from the shower maximum deeper into the calorimeter. Figure 2.1 shows total doses and particle fluences in the calorimeters' region;

<sup>2</sup>Total dose is defined as the total absorbed energy per unit mass. Although the S.I. unit for total dose is the *Gray* (Gy), where  $1 \text{ Gy} = 1 \text{ J/Kg}$ , in the high energy physics community the old unit *rad* is still used:  $1 \text{ rad} = 10^{-2} \text{ Gy} = 100 \text{ erg/g}$ .

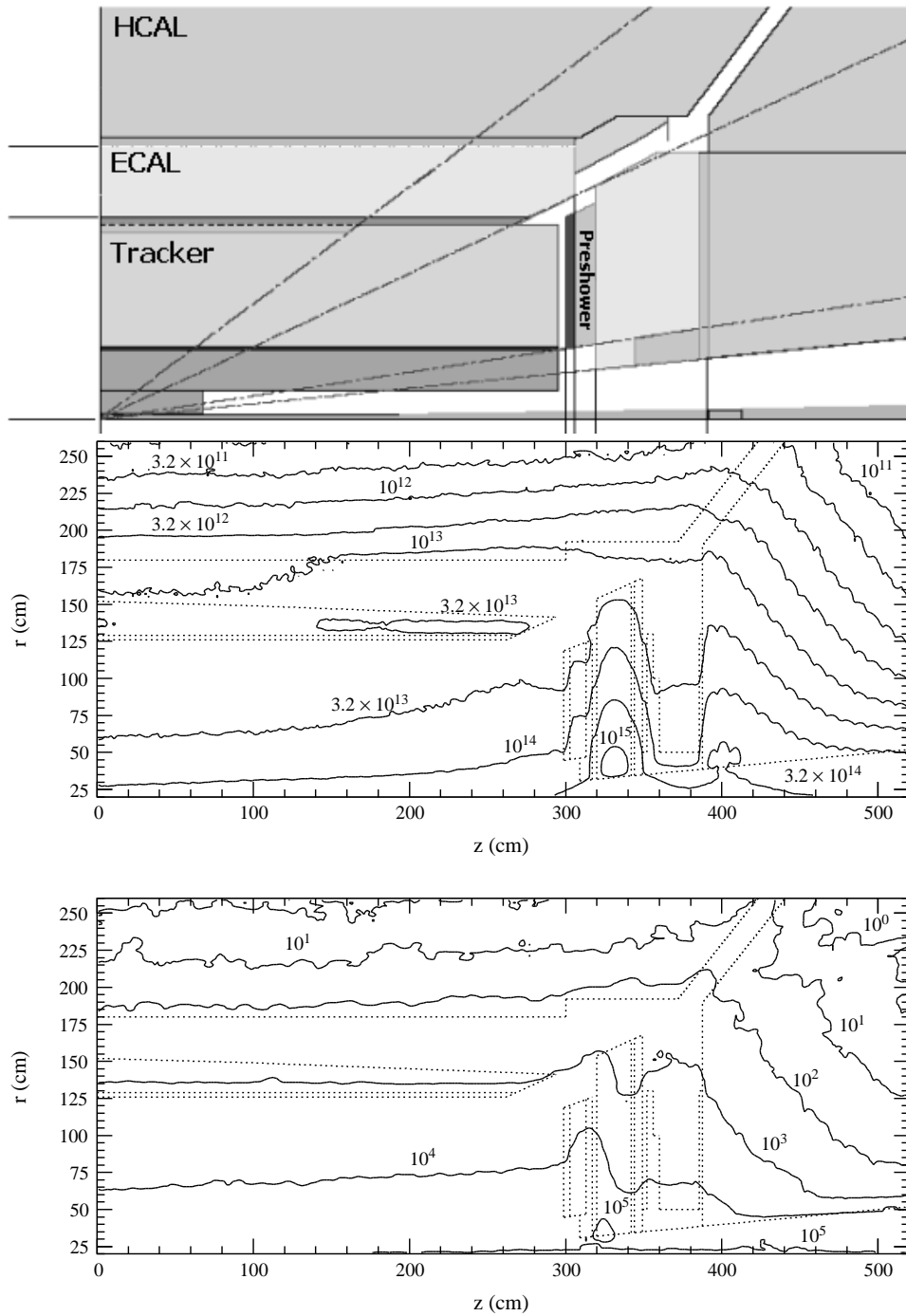


Figure 2.1: Fluence of neutrons (with energy above 100 keV) and charged hadrons in  $\text{cm}^{-2}$  (upper plot) and radiation dose in Gy (lower plot), in the region inside the solenoid [8]. The dotted lines in the graphs indicate the geometry shown above.

it also clearly demonstrates that the ECAL's crystals are the most intense source of fast neutrons inside the CMS.

Without any moderators separating it from the crystals, the preshower would be directly exposed to that neutron flux. The most effective shielding method is based on elastic scattering of the neutrons from hydrogen nuclei. Studies have shown that a few centimeters of polyethylene are sufficient to lower the energy of the neutrons below the 100 keV limit. Thus, sandwiching the preshower between two 4 cm layers of polyethylene, this fluence can be reduced by a factor of 3 [13].

An even better protection would be obtained by using a larger moderator but because of space limitations, the thickness of the moderator is crucial.

The preshower still receives a significant amount of radiation (about 3 Mrad in 10 years): this rules out every possibility of using commercial solutions for its design.

### 2.1.3 Radiation tolerant ICs

It is then clear that the integrated circuits used for the front-end<sup>3</sup> electronics of the detectors must be resistant to radiation. The need of these kind of circuits for the various applications mentioned above led, in the past, to the development of special technologies, called "radiation hardened", where particular processing methods are used in order to improve their radiation tolerance. Modifying the *process* steps is one of the three ways to improve the radiation tolerance of an integrated circuit. The two other possibilities are to use special *layout* techniques or special *circuit and system* architectures.

#### *The gate oxide*

In a metal-oxide-semiconductor (MOS) transistor, the part most sensitive to radiation effects is the gate oxide. One way to reduce those effects is to reduce the gate's thickness, which is the natural trend in modern technologies. The market of memories, microprocessors and, in general, digital integrated circuits, has driven a very fast technological evolution in the past 20 years, which has led to today's deep submicron devices with less than 2 nm gate oxide thickness.

#### *Commercial vs rad-hard*

This suggests the possibility of using modern commercial CMOS (complementary MOS) technologies in radiation environment without introducing or modifying any particular process step. Hardening a technology by introducing special processing steps is generally not convenient for us since the foundries would not modify their processes for such a small market without increasing considerably the prices.

---

<sup>3</sup>Usually, in the CERN experiments context, the electronics inside the experiments is referred as *front-end* (FE). The detectors and their immediately close analog equipment is instead called *very front-end* (VFE).

Having a radiation tolerant gate oxide does not resolve all the possible problems when irradiating an integrated circuit made in a standard deep submicron technology. To solve these problems one can still adapt the layout and the architecture of the circuits and of the system.

The use of deep submicron CMOS technologies has several beneficial aspects, such as speed, reduced power consumption, high level of integration, high volume production. Moreover commercial technologies do not suffer from the problems of radiation hardened technologies, which are more expensive and less advanced (usually a couple of generations behind). Last but not least, the availability of some radiation hardened technologies in the future is not certain, and cases have already been experienced of foundries stopping the production of their radiation hardened processes due to the drop of demand.

For this reason, in 1996, CERN's microelectronics group started to investigate the possibility of using a commercial CMOS technology to integrate the circuits to be used in the detectors. The very promising results obtained led, at the end of the same year, to the proposal of a Research and Development project<sup>4</sup> which was approved in March 1997. The aim of the project was to assess the improved radiation tolerance of submicron CMOS technologies and to study the use of design and layout techniques to increase it further. At that time, 0.7  $\mu\text{m}$  technology was the state of the art, but since then the evolution has been followed characterizing 0.5, 0.35 and 0.25  $\mu\text{m}$  technologies.

*CERN RD49*

As confirmed in the RD49 status reports [16, 17], the results were very successful, and allowed the design of integrated circuits which could stand doses of 30 Mrad and beyond [27, 15]. At the present time a 0.13  $\mu\text{m}$  technology is being studied, while a rich 0.25  $\mu\text{m}$  digital library is commonly used for design.

## 2.2 Radiation effects

### 2.2.1 Radiation effects on matter

The manner in which radiation interacts with solid materials depends on many factors, but the three main criteria of classification are charge, mass and energy of the incident particle. Protons and electrons are charged particles, while neutrons and photons are neutral particles. From the mass point of view, instead, protons and neutrons are heavy particles, while electrons are light particles.

**Charged particles** interact through the Coloumb force with the target

---

<sup>4</sup>“CERN RD49 – Study of the radiation tolerance of ICs for LHC”

material atoms inducing ionization or atomic excitation. Neutral particles instead do not exhibit this kind of behavior.

**Massive particles** can collide with the nuclei of the target material causing displacement, excitation or nuclear reactions if the energy is enough.

**Electrons** also generate Bremsstrahlung (X-rays) when decelerating into the target.

**Photons** have zero mass and no charge, therefore they have a special behavior with respect to other particles. They can interact, ordered by energy of the photon:

- by photoelectric effect, in which an electron of the target atom changes energy state, possibly ionizing the atom, and the photon is completely absorbed;
- by Compton effect, in which an electron of the target atom is set free and a residual photon is emitted;
- by electron-positron pair creation (above 1.024 MeV).

*Semiconductors and insulators*

In practice, the effects of radiation on the materials involved in microelectronic devices production can be grouped in two classes: ionization effects and nuclear displacement [5].

**Ionization** creates electron-hole pairs. The number of pairs created is directly proportional to the total absorbed dose. For this reason, the studies on the effects of ionization refer only to this quantity and not to the type of particle chosen.

**Displacement** gives origin to crystal defects, most of which are Frenkel pairs. In  $\text{SiO}_2$  at room temperature, 90% of the Frenkel pairs recombine within a minute after the end of irradiation. MOS transistors are almost entirely insensitive to displacement damage, since they are device whose conduction is based on the flow of majority carriers below the silicon-oxide interface, a region which does not extend deeply in the bulk. This phenomenon has therefore a limited importance.

### 2.2.2 Radiation effects on electrical parameters of MOS transistors

*Positive charge trapped in  $\text{SiO}_2$*

As mentioned above, MOS transistors are more sensitive to ionization than to displacement damage. In the gate (metal or polysilicon) and in the substrate the electron-hole pairs generated quickly disappear, since these are materials with small resistance. On the other side, in the oxide, which is



an insulator, electrons and holes have a different behavior, as their mobility differ by  $10^5$  to  $10^{10}$  times<sup>5</sup>.

Only a fraction of the induced electron-hole pairs will recombine immediately after being generated, while the rest will be separated by the electric field. In the case of a positive bias applied to the gate, the electrons drift to the gate electrode in a very short time whereas the holes move towards the Si-SiO<sub>2</sub> interface with a very different slower transport phenomenon<sup>6</sup>. Then, close to the interface, but still in the oxide, some holes may be trapped, giving origin to a fixed positive oxide charge  $Q_{ox}$ .

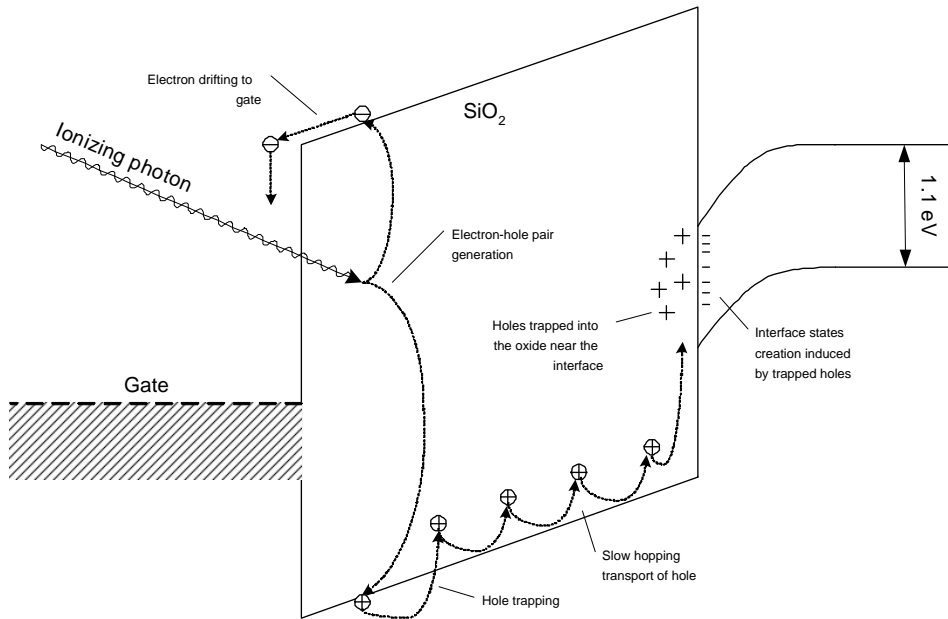


Figure 2.2: Band diagram showing the transport and trapping of holes in the oxide.

The amount of trapped charge is proportional to the number of defects in the silicon dioxide: depending on the oxide quality and on the electric field, the fraction of trapped holes varies from 1% to 100% [3, 1]. The non trapped holes which reach the interface, will recombine with electrons coming from the silicon. Moreover these electrons may tunnel from the silicon surface into the oxide and recombine with trapped holes, giving origin to a *tunnel-effect-based annealing* [25]. This effect makes the trapped charge quantity to vary with the absorbed dose rate and its history.

<sup>5</sup>Typical SiO<sub>2</sub> electron mobility at room temperature is  $20 \text{ cm}^2\text{V}^{-1}\text{s}^{-1}$ , while for holes it depends strongly on the temperature and on the electric field, and ranges between  $10^{-4}$ – $10^{-11} \text{ cm}^2\text{V}^{-1}\text{s}^{-1}$ .

<sup>6</sup>The transport of holes in SiO<sub>2</sub> is based on the concept of *small polaron hopping* [4, 24], which will not be discussed in this thesis.

The positive oxide charge lowers the threshold voltage  $V_T$  in n-channel transistors, since it attracts more electrons to form the silicon inversion. In p-channel transistors the threshold voltage absolute value is increased, or, in other words,  $V_T$  is more negative.

*Radiation induced traps at the Si-SiO<sub>2</sub> interface*

Ionizing radiation also induces the creation of interface traps. These traps have an energy laying in the silicon energy gap. Experiments indicate that the major part of the traps present above midgap are acceptors, while traps below are donors<sup>7</sup> [29, 1]. Filling those states gives rise to a interface trapped charge  $Q_{it}$ .

For this reason, in both p- and n-channel MOS transistors, the threshold increases (in absolute value), after irradiation, due to the creation of new interface traps<sup>8</sup>. Again, radiation induced trap generation is strongly dependent on the processing steps of MOS devices. Thus one of the fundamental steps for the fabrication of radiation hardened devices is the control of the gate oxide quality.

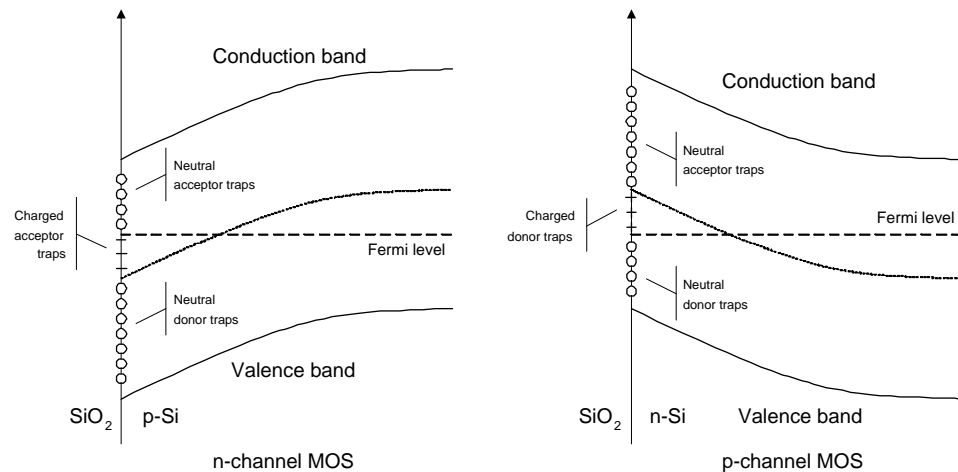


Figure 2.3: Band diagram showing the behavior of interface states for an n-channel and a p-channel transistors. The gate bias is positive for the n-channel while negative for the p-channel.

<sup>7</sup>A donor trap releases an electron when it passes from below to above the Fermi level. Donor traps are neutral when full and positively charged when empty. An acceptor trap captures an electron when it passes from above to below the Fermi level. Acceptor traps are neutral when empty, negatively charged when full.

<sup>8</sup>Considering an n-channel MOS transistor working in inversion, the acceptor traps in the upper part of the gap, being below the Fermi level, will be filled by electrons and then negatively charged, making necessary an higher gate voltage to have the same channel inversion.

The two phenomena described above cause the threshold voltage to vary with irradiation. While p-channel transistors experience only an increase of  $V_T$ , in n-channel transistors it can decrease, increase, or even be stable, depending of which is the major effect between the positive oxide charge and interface traps. Moreover the  $Q_{ox}$  is influenced by the thickness of the oxide and the dose rate: in oxides thinner than 7 nm, at low dose rates,  $Q_{ox}$  is in general negligible in respect to  $Q_{it}$ .

Thus, in modern technologies, like 0.25  $\mu\text{m}$  and below, where the gate oxide is thin, the threshold voltage shows only an increase with irradiation [1], while in the past technologies it had more complex behavior related to the balance between  $Q_{it}$  and  $Q_{ox}$ . The absolute value  $V_T$  increase, as Figure 2.4 shows for a 0.25  $\mu\text{m}$  technology, is anyway less than 80 mV after 30 Mrad irradiation.

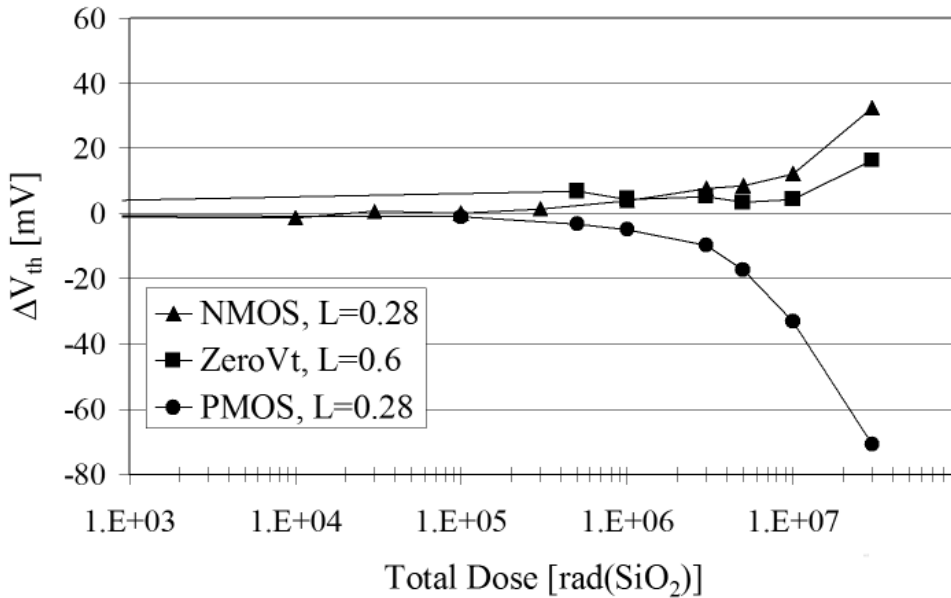


Figure 2.4: Threshold voltage shift of enclosed NMOS, enclosed Zero- $V_T$  NMOS, and normal PMOS transistors in 0.25  $\mu\text{m}$  technology as a function of the total dose [10].

In MOS devices, a thick oxide is used to isolate between different devices and, within the same device, between the source and the drain [21]. Usually the first is referred to as field oxide, while the second as lateral oxide. In many technologies these oxides are made in the same process step, like, for example, the Local Oxidation of Silicon (LOCOS). In deep submicron

processes, the thick oxide is often made with the Shallow Trench Isolation (STI) technique, that guarantees a better quality than the LOCOS.

Since the lateral oxide is much thicker than the gate oxide, it suffers more of radiation-induced positive trapped charge. This can form a parasitic path near the gate's sides connecting the drain to the source, increasing, in practice, the leakage current. As mentioned before, positive  $Q_{ox}$  lowers the threshold only in n-channel transistors, thus only in those transistors a post-irradiation leakage current is observed. In a  $0.25\ \mu\text{m}$  technology this current can grow up to the order of  $1\ \mu\text{A}$  after 10 Mrad irradiation, an unsuitable value for the fabrication of any chip! As Figure 2.5 shows, this technology can be used without any special layout technique up to 200 krad, but not

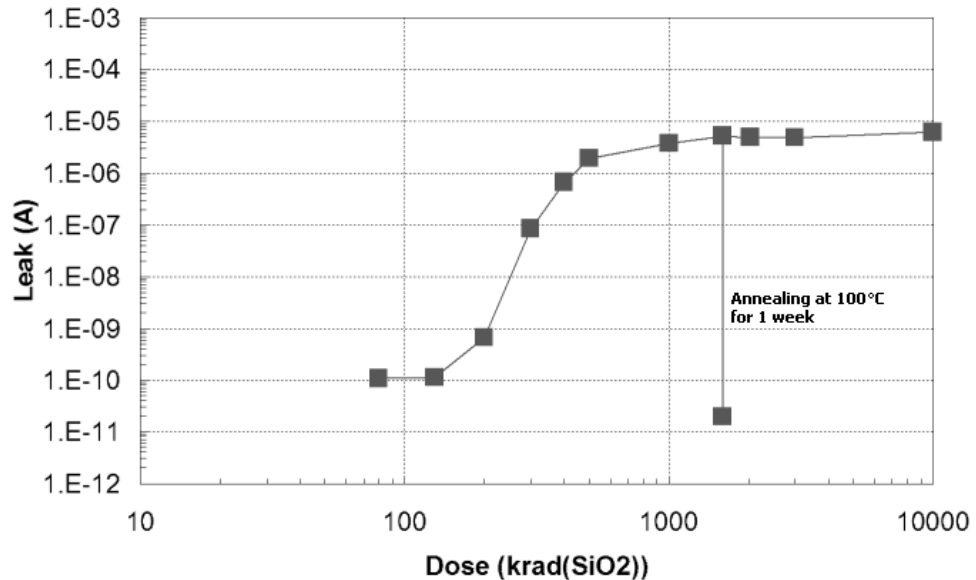


Figure 2.5: Leakage current for normal devices in  $0.25\ \mu\text{m}$  technology. The measurement was taken with  $V_{DS} = V_{DD}$ . [2].

### 2.2.3 Single Event Effects (SEE)

Single event effects are phenomena generated by one single highly energetic particle passing through a device.

#### *Single Event Latch-up (SEL)*

Latch-up is a destructive effect which can occur because of the parasitic thyristor formed by the complex junction structure built in every CMOS IC. This phenomenon is usually avoided with process and layout techniques, like for example placing well contacts very close to the devices' source. Even though, it can happen that a ionizing energetic particle passing through the

device deposits charge inside the parasitic thyristor, causing it to turn on. This effect is called “single event latch-up” (SEL). Its importance is limited in deep-submicron technologies since the presence of trench isolation between wells deteriorates the parasitic thyristor.

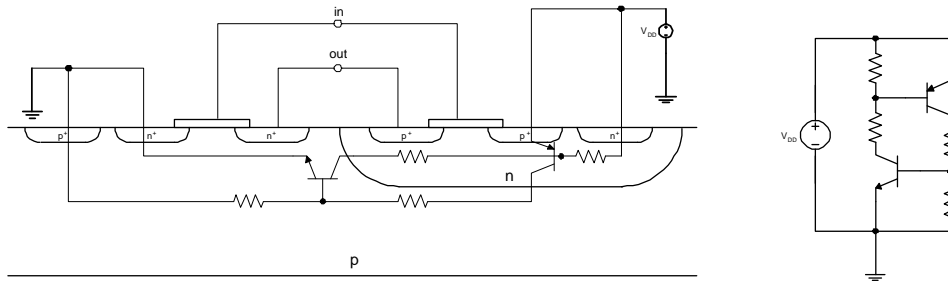


Figure 2.6: Cross-section of a CMOS inverter showing the parasitic thyristor (left) and its circuit (right).

Ionizing particles can also change the state of a circuit node and cause false information to be stored inside: this phenomenon is called “single event upset” (SEU). Unlike many other radiation-induced effects, SEU sensitivity increases with the scaling down of VLSI technologies: in fact, the minimum charge collection needed to generate the upset is proportional to the node capacitance and the supply voltage. From the circuit level point of view, dynamic logic is more sensible to SEU than static logic. SEU are dangerous in registers and memories, where the data content can be unrecoverably changed. Data redundancy is therefore needed in radiation environment applications.

Studies on SEU sensibility of D flip-flops [11], made in the 0.25  $\mu\text{m}$  technology in use at CERN, have demonstrated that, under the CMS outer tracker’s conditions, the expected average SEU rate ranges between  $2.5 \cdot 10^{-11}$  –  $9.5 \cdot 10^{-11}$  errors/(cell  $\cdot$  s). The preshower’s corresponding values are very close to the outer tracker ones, since these two sub-detectors are adjacent.

*Single Event Upset  
(SEU)*

## 2.3 Hardening against radiation

The choice of using a deep submicron technology guarantees itself a radiation hardened gate oxide. What is therefore necessary is to solve the problems related to the n-channel devices’ field and lateral oxide degradation after irradiation.

### 2.3.1 Layout techniques

#### *Enclosed Layout Transistors (ELTs)*

The primary problem which has to be addressed is the leakage current inside n-channel devices. The solution adopted in CERN's microelectronics group is to use "enclosed layout transistors" (ELTs, also called edgeless). As shown in Figure 2.7, in this case the parasitic path between the source and the drain is eliminated, as well as the lateral oxide.

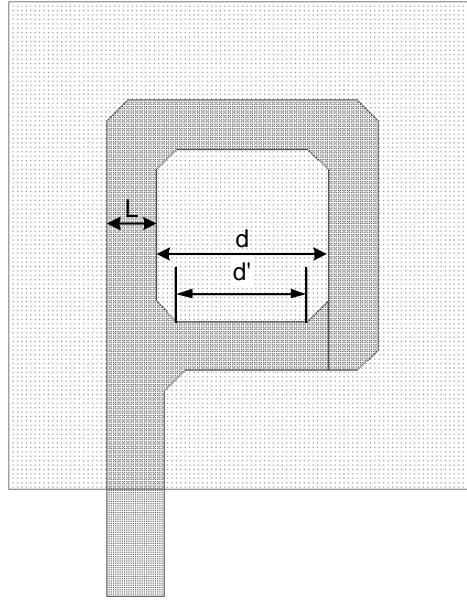


Figure 2.7: Enclosed Layout Transistor. The drain is conventionally in the center while the source is outside the circular gate.

The major disadvantages of this layout style are larger area and increase in capacitances. Moreover, the choice on the  $W/L$  ratio is limited, since  $W$  has to be enough big to allow the inner active contact to be placed.

ELTs have been used in the early days of CMOS [9] and their effectiveness in preventing leakage currents in irradiated integrated circuits is well known. Their intensive use in CERN's applications lead to the investigation of many issues important for a designer, such as modelling the effective  $W/L$  ratio<sup>9</sup>. There is a wide range of possible enclosed shapes: squared,

<sup>9</sup>As described in [12], the model for the effective  $W/L$  of enclosed transistor, if applied to the shape in Figure 2.7, leads to the following expression for the aspect ratio:

$$\left(\frac{W}{L}\right)_{eff} = 4 \frac{2\alpha}{\ln \frac{d'}{d' - 2\alpha L_{eff}}} + 2K \frac{1 - \alpha}{1.13 \cdot \ln \frac{1}{\alpha}} + 3 \frac{d - d'}{2L_{eff}}$$

where  $\alpha$  is constant usually set to 0.05, while  $K = 7/2$  for short channel transistors ( $L \leq 0.5 \mu\text{m}$ ), otherwise  $K = 4$ . To derive this expression, the enclosed transistor is

octagonal, squared with corners cut at 45 degrees and all of them can have a different behavior and require a separate model. To simplify the problem, one specific shape was chosen, compatible with the design rules of the process: square with corners cut at 45 degrees so that the size of the cut is constant for all the gate lengths (see Figure 2.7).

The second problem which can be solved with a layout technique is the leakage between different devices [1]. This is done surrounding each n-channel device with a p+ guard ring. This method has been verified to be very effective but the drawback is again the big consumed area. Moreover, guard rings avoid the generation of SEL by lowering the gain of the parasitic NPN bipolar transistor.

*Guard rings*

### 2.3.2 Circuit and system techniques

While designing circuits for radiation environment applications, one must take into account and foresee the drift of the circuit's operating point due to absorbed total dose. For digital circuits, the synchronous mode of operation limits the sensitivity to electrical parameters' variation [1].

*Synchronous operation*

SEU tolerance has also to be implemented at this level of hierarchy: as mentioned before, static logic is indeed less sensible to SEU, but its use is not enough to guarantee immunity. In fact, SRAM and flip-flop contents can be changed and have to be protected if crucial for the application.

*Static logic & data redundancy*

Of course data redundancy is more important in state machines, where a wrong stored information can be very harmful, than in data paths. An error in a control logic state machine can damage the whole system behavior, while in data paths it is usually confined to the corrupted data segment. For this reason, state machines are *triplicated* in radiation environment applications, and therefore in this work.

*State machines triplication*

Triplication can be done in a few ways. Three of them are shown in Figures 2.9, 2.10, 2.11, while Figure 2.8 resembles a standard state machine.

State machines are usually composed by a combinatorial logic block and a set of flip-flops or registers. The more general case of Mealy state machine only will be analyzed, since Moore machines are a subset of the others<sup>10</sup>. These memory devices store the current state vector, while the logic evaluates the next clock cycle's state and the outputs.

---

decomposed into three parts. The first corresponds to the linear edges of the transistor, the second to the corners without the 45 degrees cut, which then is taken into account in the third part. It can be shown that the minimum reachable aspect ratio is around 2.26 with this geometry.

<sup>10</sup>In Moore state machines the output depends only on the state vector. In Mealy machines the output depends on the state vector and on the input vector.

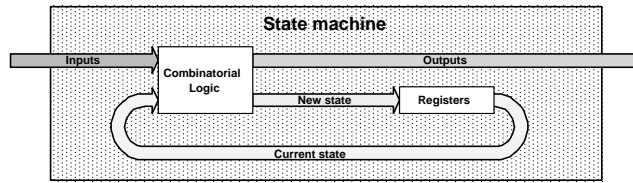


Figure 2.8: Standard state machine.

In triplicated state machines a new block is necessary: a so called “majority voter”, purely combinatorial. In our case, it has three inputs and one output which is 1 if at least two of the inputs are 1, while it is 0 if at least two of the inputs are 0. The voter’s output is never undefined, as can be seen from the truth table in Table 2.2.

I2	I1	I0	Out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Table 2.2: 3-Input majority voter truth table.

Since a state machine is repeated three times, one or more majority voters take care of deciding the right state and output when a disagreement takes place. In principle, the three machines could run independently with a voter connected to their output, as shown in 2.9.

This configuration is not suitable for application requiring a long operation time without reset: after a SEU occurred in one of the state machines, a second SEU in another machine will make the output to be wrong. This can be avoided if the state machines evaluate the voted state instead of their own stored state.

In order to do that, the stored state loop back to the combinatorial logic has to be broken in each state machine, creating an *open state machine* with state vector input and output. Then, the voter must be placed to connect them: three registers will drive the voter, which will decide the correct current state and send it to the combinatorial parts. Figure 2.10 shows the simplest triplication scheme with feedback of the voted state.

A problem arises from this second approach: a SEU on the voter output



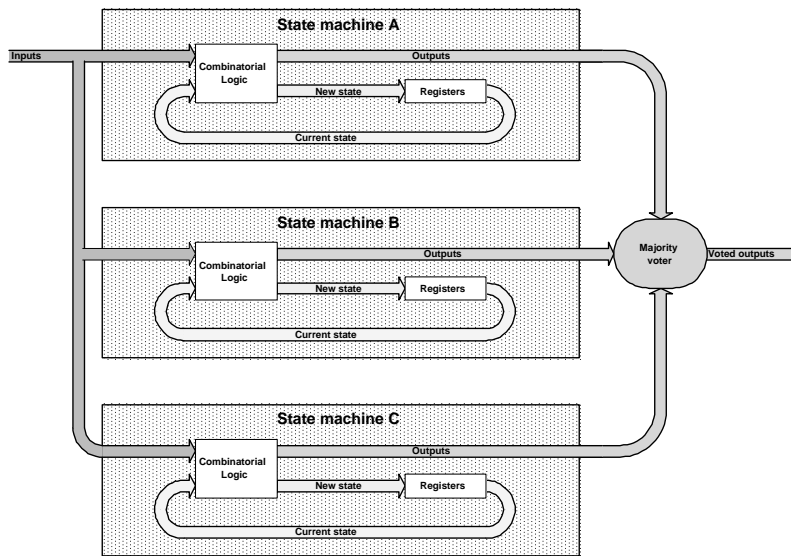


Figure 2.9: Triplicated state machine with no feedback.

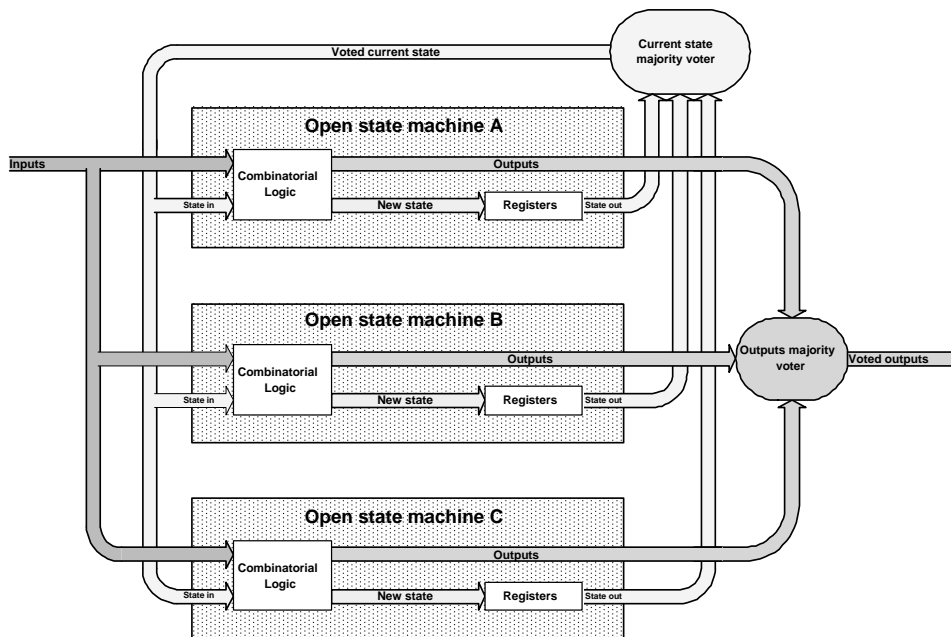


Figure 2.10: Triplicated state machine with shared feedback.

could cause a bad state vector to be loaded in all the three machines at the same time! Even if this is very unlikely, to resolve it, one can let the voted state vector feedback to be given only to the first state machine, and

then make it propagate down to the other two in the next 2 clock cycles. Figure 2.11 shows this new solution.

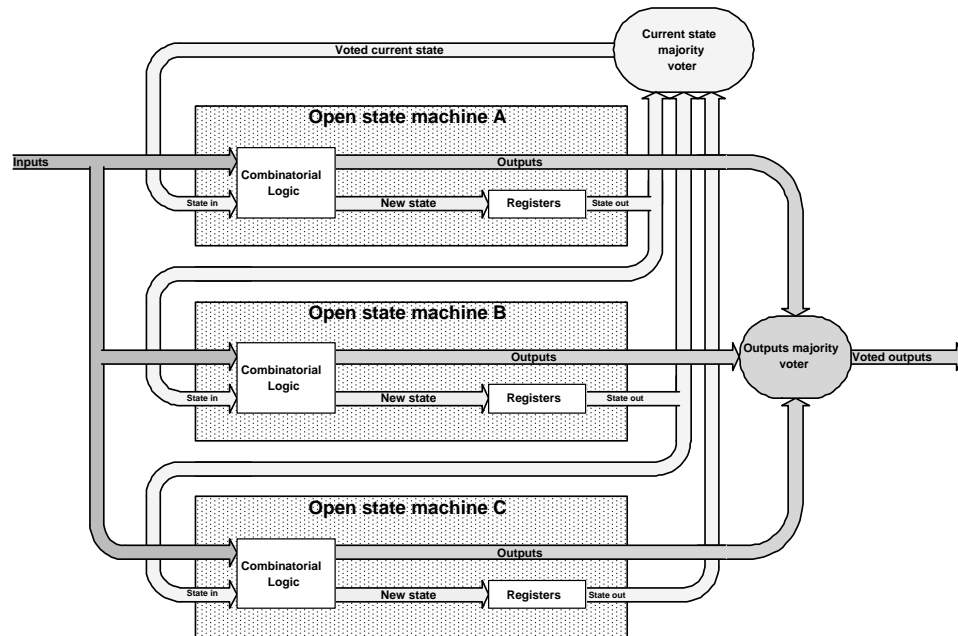


Figure 2.11: Triplicated state machine with state propagation.

As can be seen, the second state machine's input state vector is the previous machine's output state vector, and the same happens for the third machine. If no more than one SEU occur within three clock cycles, an eventual bad state vector will propagate through the state machine chain and die in the last one, while the correct state vector will always be restored in the first.

This last one is the solution adopted in this work during the design of all the state machines.

### *Clock nets' SEU*

An especially harmful kind of SEU can happen on the clock nets. In that case two false digital signal transitions are inserted where they shouldn't be. Flip-flops connected to that net will sample the input data in the wrong moment, corrupting their content. Anyway clock nets have usually a large capacitance and a powerful driver connected, thus they are intrinsically immune to SEUs.

## 2.4 A radiation tolerant digital standard cells library

In order to help in the design of complex digital ICs, a digital library has been designed and tested in a  $0.25\ \mu\text{m}$  technology [23, 19], while a new library in  $0.13\ \mu\text{m}$  technology is under development. Both these two libraries exploit radiation hardening techniques.

### 2.4.1 The CMOS $0.25\ \mu\text{m}$ library

The basic features of the technology are given in Table 2.3.

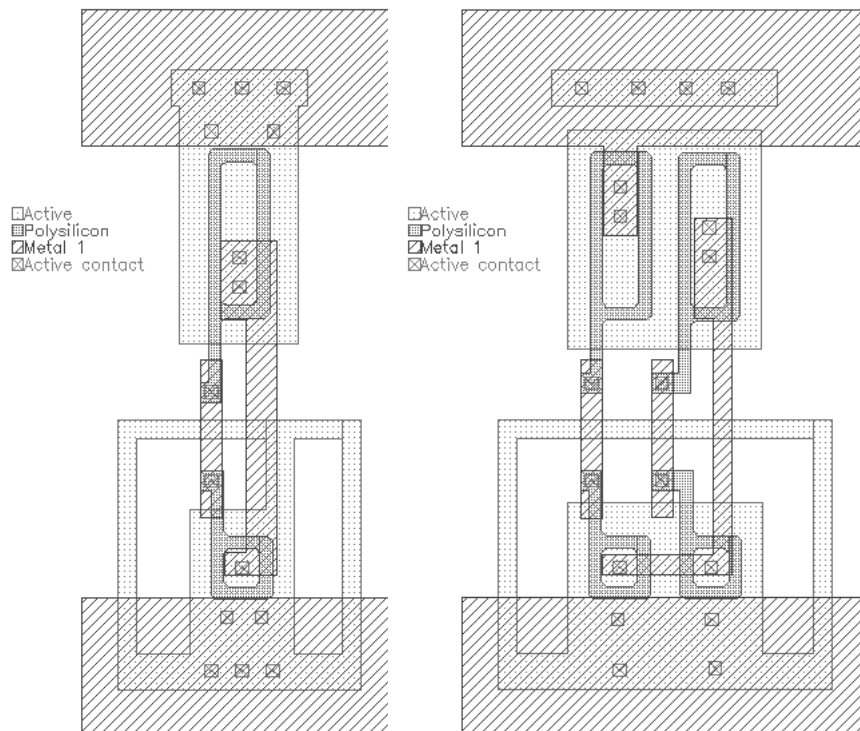


Figure 2.12: Inverter gate (left) and 2-input NOR gate (right).

The standard cells are designed to be abutted one to the other in horizontal rows. Figure 2.12 shows two library cells.

The power rails are routed in the first metal layer horizontally all along the rows; great effort was spent to keep intracell interconnections on the first metal layer, leaving the rest of the metal layers for global routing. For that purpose the salicided polysilicon layer was used as a local intracell interconnect, but since polysilicon cannot be allowed to cross the guardrings, this layer was used only for horizontal routing.

Minimum lithography	0.24 $\mu\text{m}$
$L_{eff}$	0.18 $\mu\text{m}$
$V_{DD}$	2.5 V
Gate oxide thickness	5.0 nm
Process	Twin well CMOS
Device isolation	Shallow trench (STI)
Ti salicidation	On $n^+$ and $p^+$ polysilicon and diffusions
Interconnectivity	2 to 5 metal layers

Table 2.3: 0.25  $\mu\text{m}$  technology features.

The area penalty paid for ELT style and the guardrings is anyway mitigated by the small feature size of the technology: the only alternative to this approach would be to use process radiation hardened technologies which offer overall a much smaller device density.

The library contains combinatorial logic gates, like NANDs and NORs, as well as flip-flops and latches. A set of I/O pads is also available.

# Chapter 3

## The Kchip

The primary objective of this thesis has been to develop a digital radiation-tolerant ASIC for data readout from the CMS preshower, which is called the “Kchip”. In the introduction, the goal and the basic concepts of the CMS preshower have been explained. In the next sections the electronic equipment concerning the preshower front-end (FE) system will be described with more detail. Later in this chapter the design requirements needed, the techniques employed and the internal operation of the Kchip will be exposed.

### 3.1 The CMS preshower front-end system

The front-end readout system [18] is composed by a few units, cascaded one to the other, which, following the data path from the source to the outgoing optical fiber, are (see Figure 3.1):

- The silicon strips detector;
- The PACE chipset, composed by a charge-sensitive amplifier and a pipeline analog memory;
- The analog-to-digital converter (ADC);
- The Kchip;
- The Gigabit Optical Link (GOL) chip, which is a high-speed data serializer and laser driver;
- The light emitting diode laser;

A detector, together with a PACE chipset and a Detector Control Unit (DCU), form a micromodule (see the Introduction). Up to 10 micromodules are then connected to one motherboard made out of an ADC stage, a number of Kchips, one GOL and one laser diode per Kchip, plus some control logic. In practice every Kchip can read the data coming from up to four detectors.

The 800 Mbit/s optical fiber goes then into what is called the *counting room*, where event building and Level-1 trigger evaluation take place. Two pairs of optical fibers are also connected to a motherboard for slow control signaling and clock distribution. All the communication between the motherboard and the counting room are implemented with optical fibers, while with wires among the motherboards.

### 3.1.1 The control logic

The control logic comprises the Communication and Control Unit module (CCU), the Phase Locked Loop (PLL) and other smaller components like the Digital Optical Hybrid (DOH).

#### *The PLL*

The PLL device takes care of receiving and extracting the 40.08 MHz LHC clock signal (CLK), which comes from the counting room through an optical fiber token ring, and distributing it to the rest of the motherboard. It also performs clock jitter reduction and programmable deskewing with steps of 1.04 ns. Along with the clock, the Level-1 trigger is transmitted, coded in missing pulses, and the PLL provides its extraction. Figure 3.2 shows the Level-1 trigger T1 decoding. In addition, the trigger channel carries other coded information. All these signals are referred as *fast timing control signals* and will be clarified later.

#### *The DOH & the token rings*

The DOH is responsible of the electrical to optical (and viceversa) translation. Fast switching electrical signals are often implemented in Low Voltage Differential Signaling (LVDS) standard<sup>1</sup>, a current-steering mechanism which provides low power and very low noise necessary for the FE analog parts.

A primary and a secondary token ring are both available for slow control bidirectional communication. A DOH is mounted on each one of the two master motherboards, each one serving one token ring. The token ring involves up to 12 motherboards. Having two rings provides redundancy for fault tolerance: the networks are doubled and cabled in such a way that, if any motherboard fails, there will still be a path to reach the other ones.

The two DOHs in particular take care of generating the asynchronous reset signals (RESET1b and RESET2b): when no light is observed in the control data fiber a reset is issued. The two low-active signals are the OR-ed inside the CCU module to obtain a single RESETb.

#### *The CCU*

The 40 Mbit/s fiber channel token rings are connected to the CCU, that provides slave control of the connection. In this way, the various circuits in the front-end can be accessed through this component, by a set of 16 lines

---

<sup>1</sup>ANSI/TIA/EIA-644(LVDS)

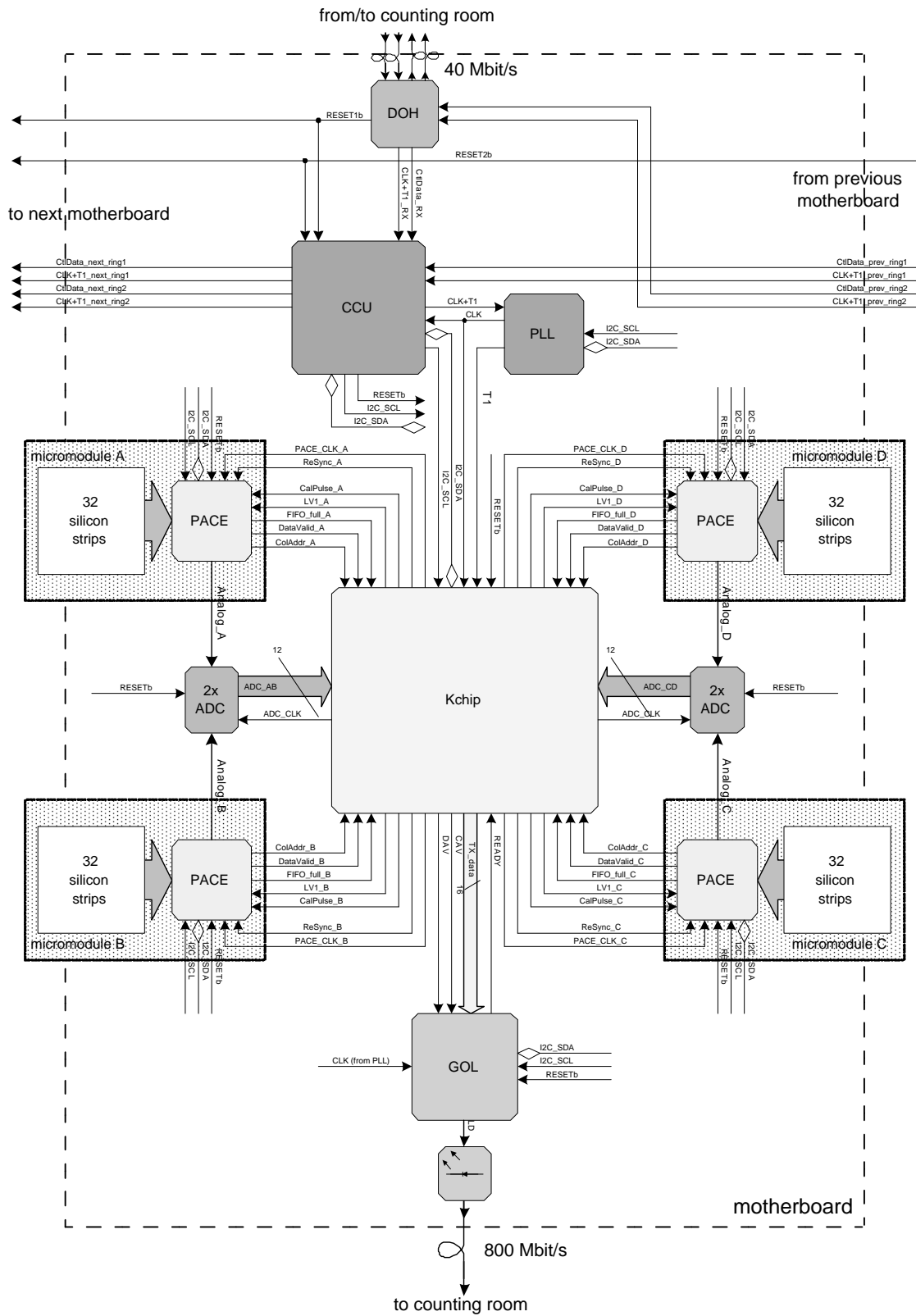


Figure 3.1: Simplified master motherboard schematic showing the preshower FE system architecture.

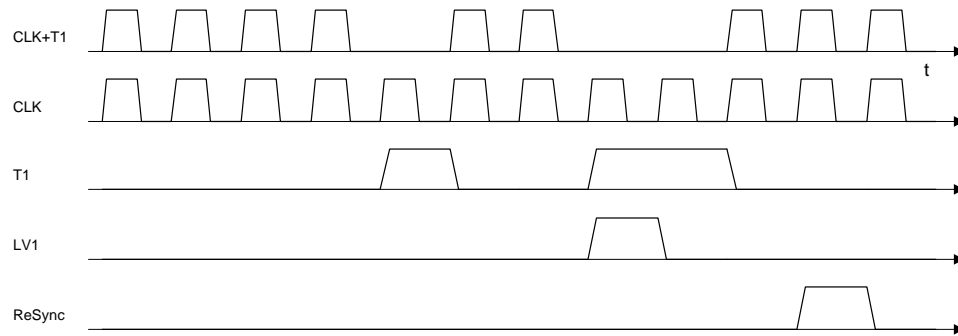


Figure 3.2: Level-1 trigger decoding from the clock signal and example of fast timing control signals derived.

of I<sup>2</sup>C interface, which is a standard serial communication protocol created by Philips. The FE system is then capable to be programmed and verified from the outside.

### The DCU

The DCU (not shown in the schematic) is dedicated to the measurement of environmental parameters of the system (temperatures, voltages, leakage currents and so on).

### 3.1.2 Fast timing control signals

Inside the Level-1 trigger (T1) four possible commands are coded. These commands are represented, in the T1 stream, by a triplet of bits beginning with 1. Table 3.1 shows the four possible commands. Back-to-back commands are allowed in this scheme.

T1 Pattern	Command	Description
100	LV1	Readout request
111	CalPulse	Calibration request
110	ReSync	Resynchronization request
101	BC0	Bunch crossing zero reference

Table 3.1: The four possible fast timing control commands coded into the Level-1 trigger.

Even though, this structure prevents two commands to be sent in the same time or within less than 3 clock cycles, and this is especially true for the readout request LV1. Therefore, when a readout request arrives, the system has to send the data relative to at least 3 clock cycles.

The ReSync command is sent in order to clear all the front-end data pipelines and error conditions, and is, in practice, a synchronous reset signal



destined to the readout logic only. Error conditions can be created by SEUs inside the state machines present in the system.

Since a bunch crossing counter is kept as reference for the event builder, the BC0 command serves to set this counter to 0 in the whole detector.

Eventually, the CalPulse signal is needed to perform a calibration of the FE analog parts.

### 3.1.3 The silicon detector

The  $63 \times 63 \text{ mm}^2$  silicon detector, is made of 32  $p^+$  strips on n bulk, with a 1.9 mm pitch and 61 mm length. In order to collect the deposited charge by a particle passing through each strip is kept reverse biased. A strip has an estimated capacitance of about 40 pF. The physics performance of the system requires the charge deposited on the strips to be measured with an system overall  $\approx 5\%$  accuracy.

The signals coming from the strips have a  $\approx 12$  bit dynamic range, which is very large. This value prevents any analog transmission to the counting room, since an analog optical link with such a dynamic range has not yet been proven. Moreover, no zero suppression is performed in the front-end because no fully lossless and efficient algorithm has been found yet. These two last statements explain the requirement of an high speed data link.

The slope of the signal generated by a hitting particle lasts usually more than one clock cycle: three consequent samples are therefore taken at every readout request.

### 3.1.4 The PACE chipset

The silicon detector is DC coupled with the PACE chipset. The chipset is composed by two ASICs: the Delta chip and the PACE3 chip. Figure 3.3 shows the PACE chipset internal structure<sup>2</sup>. These ASICs are designed in the same  $0.25 \mu\text{m}$  CMOS technology as the Kchip.

The Delta chip is a high-gain low-noise charge-sensitive preamplifier which performs also analog signal shaping. The chip is able to measure the charge on 32 silicon strips in parallel. The bias conditions of the preamplifiers are programmable through a link with the PACE3 chip. The preamplified analog data is then sent to the PACE3 chip.

*Preamplification and shaping*

The PACE3 chip main goal is to sample and retain the analog data from the silicon strips in its pipeline memory while the Level-1 trigger is evaluated.

*Analog memory*

---

<sup>2</sup>The PACE chipset has been subject to several subsequent designs in different technologies. Here is reported the latest information. Up-to-date technical documents can be found in the CMS preshower webpage <http://cmsdoc.cern.ch/cms/ECAL/preshower/>.

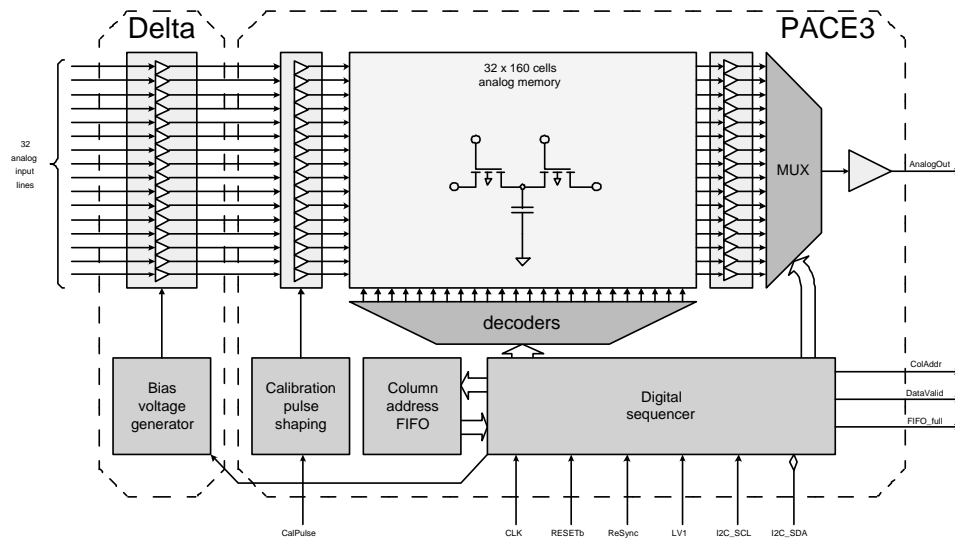


Figure 3.3: The PACE chipset internal structure.

The analog memory has therefore 32 rows able to store this data. This storage is pipeline structured: data is written in one column at every clock cycle, erasing the previous content. Since the Level-1 trigger latency is of 128 clock cycles, which means  $3.2 \mu\text{s}$ , at least 128 memory columns are needed to make sure that, when the trigger arrives, the requested data is still in the memory. Nevertheless, a major size is necessary because, whenever a trigger arrives, 3 columns are read: the one to which the trigger refers and the two following ones. In order to complete this operation correctly a size of 160 column has been chosen<sup>3</sup> and the memory control logic keeps track of all the columns requested by the trigger, making the write pointer to skip those locations, and preserving their content until read.

The  $32 \times 160$  locations analog memory is implemented with 2-transistors dual port cells and the data is stored on a metal-insulator-metal capacitance (MIMCAP).

The choice of using an analog memory instead of a digital one after an A/D conversion was done only because of the huge number of fast high-resolution ADCs needed for this last solution, which would consume lots of power and space.

#### Address FIFO

The digital control logic makes use of a 8 bit  $\times$  48 locations deep FIFO to store the addresses requested for readout. Every time a readout request

<sup>3</sup>Queueing theory studies were done to decide this value, since the Level-1 trigger readout request is considered as a Poissonian source. The same kind of studies have been carried out when designing the Kchip: these are reported later in this chapter.

arrives, the three corresponding addresses are pushed in. These are then popped out to drive the column decoder when transmissions are performed. If the FIFO gets full, no more readout requests can be accomplished until some data is sent out.

Data is delivered to the DC-coupled ADC through a multiplexer that scans all the rows one by one at half frequency. The choice of using this output rate was done to preserve the analog performance of the chip. The output delay foreseen is of 35 ns.

*Output multiplexer*

Table 3.2 describes briefly the PACE chipset interface pins which comprise also the I<sup>2</sup>C protocol signals coming from the CCU. Notice that the clock signal comes from the Kchip, and not from the PLL: the Kchip takes care of the PACE3 supervision. Moreover, as can be seen, three of the four fast timing control commands are decoded for the PACE3 by the Kchip.

*Interface signals*

Pin name	Type	Level	from/to	Description
CLK	Input	LVDS	Kchip	Sampling 40 MHz clock
RESETb	Input	CMOS 2.5 V low-active	CCU	Asynchronous reset
CalPulse	Input	LVDS	Kchip	Calibration Pulse
ReSync	Input	LVDS	Kchip	Pipeline synchronous reset
LV1	Input	LVDS	Kchip	Level-1 trigger readout request
I2C_SCL	Input	CMOS 2.5 V	CCU	I <sup>2</sup> C clock
AnalogIn	Input	analog	detector	data in, DC coupled
ColAddr	Output	LVDS	Kchip	Read column address, serialized
DataValid	Output	LVDS	Kchip	Data qualifier
FIFO_full	Output	CMOS 2.5 V high-active	Kchip	Column address FIFO full
AnalogOut	Output	analog	ADC	data out, DC coupled
I2C_SDA	Bidir.	open collector 2.5 V	CCU	I <sup>2</sup> C serial data

Table 3.2: PACE chipset interface signals.

The `CalPulse` signal is used directly in a calibration circuit which can generate and inject current pulses of programmable total charge into selected sampling channels. Channel selection is performed via I<sup>2</sup>C. The circuit uses capacitors, charged at an I<sup>2</sup>C programmable voltage, which, when the `CalPulse` is high, are connected to the channel inputs. The phase delay of this signal in respect to the sampling clock is therefore crucial since it controls the charge injection between adjacent samples. On top of that, the `CalPulse` signal should have a duration of at least 200 ns. The task of building a suitable `CalPulse` signal is left to the Kchip.

The `ColAddr` signal is a serial data line which, every time a column of the memory is read out, transmits the column's 8-bit address. This is provided as reference for PACE3 chip synchronization: since many PACE3

are present in the preshower, they should all send the same column addresses at the same time; if one of them is out of sync it can be easily be identified by its `ColAddr` signal. Figure 3.4 shows an event readout timing diagram. The column addresses are sent *before* the data.

The diagram shows also the `DataValid` signal behavior, which raises when the first column address bit is on the line, and lowers after the last analog data row is read. Since an event is composed by 3 columns, three `DataValid` pulses are present in a single readout operation. A `DataValid` pulse lasts 73 clock cycles, and between two pulses at least 19 cycles must be present. Therefore full column readout takes 92 cycles, equivalent to  $2.3 \mu\text{s}$ , while a full event readout takes 276 cycles equivalent to  $6.9 \mu\text{s}$ . This means that the maximum event readout rate is 145 kHz which is bigger than the maximum readout request (LV1) rate of 100 kHz.

Last, but not least, the `FIFO_full` signal tells the Kchip that no more readout requests (LV1 pulses) will be accepted, since the column address FIFO is full.

### 3.1.5 The AD41240 ADC

The AD41240 employed in the front-end system is a CMOS  $0.25 \mu\text{m}$ , 4 channels, 12 bit, 40 Msample/s ADC. Nevertheless, it has only two 12-bit output buses, since, for each one of them, two ADC channels are multiplexed using the Double Data Rate (DDR) technique<sup>4</sup>. The internal operation of the ADC uses 5 pipeline stages, therefore the output data has a latency of 5 clock cycles in respect to the input.

Two ADCs per Kchip are utilized in order to avoid some crosstalk between the signals, using only two of the four channels. The unused channels can be powered down individually. Even though the analog data from the PACE will arrive at half frequency, the AD41240 will be driven by the full frequency 40.08 MHz clock.

### 3.1.6 The Gigabit Optical Link chip

The GOL [26] is an high-speed multi-protocol transmitter ASIC: it was designed to support both the 8B/10B (eight-to-ten) and the Conditional-Invert Master Transition (CIMT) line coding schemes which are defined, respectively, in the IEEE 802.3 standard (1998) Ethernet, and in the Hewlett Packard's G-Link protocol [30]. Both schemes introduce an overhead of two additional bits every eight bits of data. The chip is made to receive the LHC 40.08 MHz clock as input, and is able to send either 16 or 32 bits of data per clock cycle: these values correspond to data bandwidths of 640 Mbit/s and 1.28 Gbit/s respectively, and are called slow and fast operation modes. With the coding overhead added the optical link bit rates required are therefore

---

<sup>4</sup>DDR uses both edges of the clock to transmit and sample data.

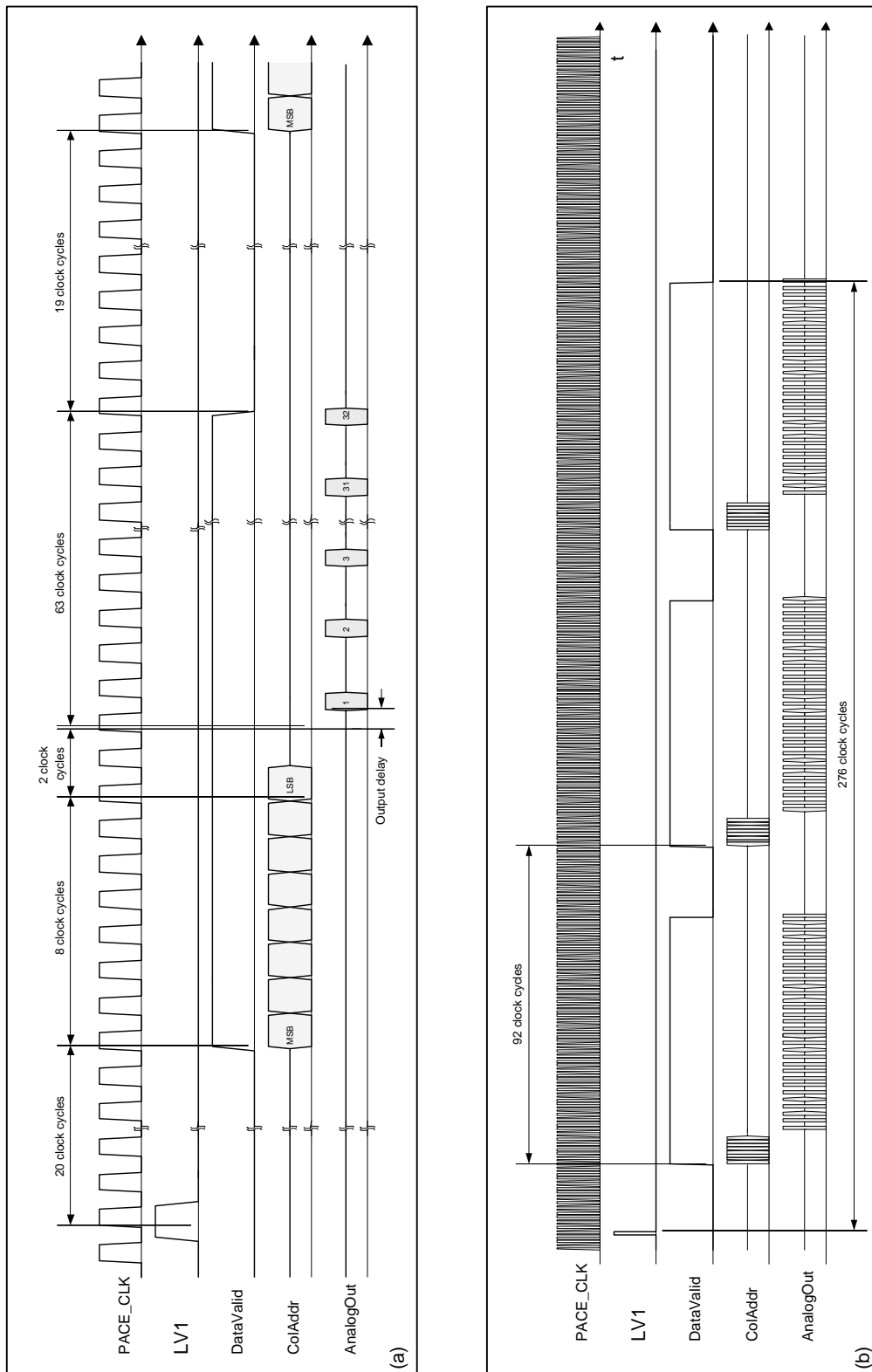


Figure 3.4: PACE3-to-Kchip interface timing diagram. In (a) a detailed description of the handshake is given, while (b) resumes a whole event readout.

800 Mbit/s and 1.6 Gbit/s, respectively. By hard-wiring two configuration pins it is possible to choose between G-Link and Ethernet, and between fast and slow operation.

The GOL can be directly connected to a laser diode since it is equipped with a programmable laser driver buffer stage.

In the preshower front-end system, the GOL will be used in slow mode, but keeping the possibility of using both the two communication standards available.

#### *Transmission control*

Basically, the data comes to the chip in 32 data lines along with two transmission control qualifiers: in case of slow operation mode, only 16 data lines will be used. At the link layer, the data is divided in *frames* that are 10-bit wide for the Ethernet mode and 20-bit wide for the G-Link. This means that in the Ethernet mode a double number of frames per clock cycle will be transmitted compared to the G-Link mode. The two transmission control pins are shared between the two communication standards: the first pin is called either *DAV* or *tx\_en*, and the second pin either *CAV* or *tx\_er*. Their behavior is explained in Table 3.3.

DAV/tx_en	CAV/tx_er	G-Link	Ethernet
0	0	Idle	Idle
0	1	Command	Carrier Extend
1	0	Data	Data
1	1	Command	Error Propagation

Table 3.3: GOL transmission control description.

As can be noticed the two protocols have more or less the same relationship with the transmission control pins, but the link commands behave differently in the two cases. In the CIMT scheme a command frame is a special combination of the data inputs, while in the Ethernet mode *CarrierExtend* and *ErrorPropagation* are the only two fixed command sequences possible.

In the G-Link mode two additional data flags can be fit into a data frame, and two different idle frames are available.

#### *Operation and timing*

The internal serializer of the GOL needs an high-frequency clock in order to work. This clock is generated internally by a PLL from the input 40.08 MHz clock. This last internal device requires a certain amount of time to lock onto the input clock, and only after that, the transmission can be performed correctly. Moreover, the PLL lock status can be lost due to SEUs or input clock imperfections. A specific signal tells the neighboring circuitry

its status: the high-active `READY` output. When `READY` is low no data will be transmitted.

### 3.1.7 The I<sup>2</sup>C interface

Many units in the front-end have configuration registers which need addressing capability from the outside through the slow control token ring. An efficient way to implement this feature is making use of the I<sup>2</sup>C bus protocol created by Philips. Here is reported a brief description of the standard.

The interface is composed by 2 electrical lines:

- a unidirectional clock line, called `I2C_SCL`;
- a bidirectional data line, called `I2C_SDA`.

The protocol is studied to let a master transceiver communicate with many slave transceivers. In our case, the CCU is always the master on the I<sup>2</sup>C bus, while the other units are slaves. The master generates the I<sup>2</sup>C clock, and decides the direction of communication.

The I<sup>2</sup>C clock line is not running when the bus is idle. A read/write operation on the slaves, also called *transaction*, is initiated by the master by putting a '0' on the data line and then starting the clock. A transaction always begins with the transmission, from the master, of an address which identifies the desired slave and register. The address bit width is variable. The addresses travel on the data line with the most significative bit first and synchronously with the I<sup>2</sup>C clock. A direction bit is appended to the address, and selects between read ('1') and write ('0').

After the reception of the address, the selected slave answers with an acknowledgement symbol on the data line. Then, either the master or the slave, depending on the chosen direction of communication put their data on the bus, synchronously with the I<sup>2</sup>C clock. Only 8-bit data transfers are allowed. The transaction ends with the master stopping the clock and then releasing the data line that goes to '1'.

The bidirectional data line is implemented with an open-collector pad technique: the input/output pads connected to this line have only a pull-down transistor, while pull-up is guaranteed by an external resistor.

The CCU provides a number of 16 I<sup>2</sup>C buses (that means clock and data pairs) to be connected to the rest of the front-end: this guarantees enough addressing space for all the units. The I<sup>2</sup>C clock runs slower than the system clock, guaranteeing data rates in the range 1 Kbit/s – 1 Mbit/s. Generally, the slaves should not rely on the I<sup>2</sup>C clock synchronicity with the system clock, therefore synchronizers must be used to avoid metastability problems.

## 3.2 The Kchip

### 3.2.1 Functionalities

From the description of the front-end electronics it can be resumed that the Kchip is demanded to:

- Collect the data from up to 4 high resolution (12-bit) ADCs at a 20 MHz rate;
- Collect and de-serialize the PACE3 column addresses;
- Provide an additive buffering of the data, minimizing the probability of lost events;
- Format the above described data to a link-suitable packet, adding data redundancy, maximizing the channel bandwidth and taking into account that two link standards can be used;
- Deliver the data packets to the GOL and perform high-level link control;
- Receive and decode the fast timing control signals from the control room;
- Take care of the connected PACE3 chips synchronization giving them the clock and checking their output signals;
- Filter readout requests that would overflow the PACE3 chip;
- Perform a first signal shaping for the calibration pulse;
- Also take care of synchronization between the ADCs and the PACE, giving the clock to the ADC as well;
- Generate a set of error messages whenever data can't be correctly read out.

Specifications include also low power consumption, low area and, not less important, testability. However, it can be said that speed is not crucial in this design, since the technology chosen is fast<sup>5</sup> and the clock is relatively slow.

The final design contains  $\approx 13000$  standard cell gates plus  $\approx 80$  kbit static memory, equivalent to  $\approx 660000$  total transistors, and is a  $6 \times 5$  mm<sup>2</sup> chip with 152 I/O pads.

---

<sup>5</sup>The typical inverter propagation delay is  $\approx 48$  ps.



### 3.2.2 Design tools and techniques

The design of a thousands-gates ASIC can't be performed without the help of Computer Aided Design tools like automatic place and route. The basic use of digital standard cells eases a lot the job, and, the employment of this technique gives to the designer the possibility of coding the whole project in a Hardware Description Language (HDL) like VHDL or Verilog.

Hardware Description Languages allow a digital circuit to be described, or modelled, using words instead of schematics, and this is done essentially in three ways, since three kinds of description are possible.

*Hardware Description  
Languages*

It is clear that a schematic can be directly translated to an HDL by declaring every component in the net and listing all the connections among them; this is called a *physical* description (or else a "netlist"). Physical descriptions are the ones which contain the maximum amount of information, but, at the same time, they are the least human-readable: a classic schematic is more readable than a netlist.

Another way to model a circuit is telling what it does and especially how it does its job: this is called a *functional* description. It is usually shorter and clearer than a physical description but it can still be translated to this last one. The action of translating a functional description to a physical one is called *synthesis*, while the opposite action is called *analysis*.

The last option is to model the circuit only by its external behavior without specifying anything of its internal functionality: this is called a *behavioral* description. This kind of description can't be translated to any one of the other two, thus its use is made only for simulation purposes. On the other hand, a behavioral description is the easiest way to model a component.

Apart from very specific cases, the choice is often to employ a functional description and then feed it to a tool called synthesizer<sup>6</sup>. Given a library of characterized standard cells and some constraints and directives, the synthesizer is able to produce a physical description which respects timing specifications and is functionally equivalent to the first model. On top of that, the synthesizer tries to optimize the circuit, by some extent, following the directives specified by the user.

*Synthesis*

Physical descriptions can be read by a Place & Route (P&R) tool, which

*Place & Route*

---

<sup>6</sup>The real border between functional and behavioral descriptions is in fact decided by what a synthesizer is able to translate and what is not. However, functional models made for synthesis are usually optimized, by a certain amount, to the interpreting capabilities of the synthesizer: trying to synthesize a circuit modelled without keeping in mind since the beginning its goal, even if possible, would probably result in a poor design in terms of area, speed and power consumption.

decides the actual position of every component on the die area and connects them together with metal tracks. The tool requires an initial *floorplanning* of the area, which means that the physical dimensions of the die, the position of the pads and the macrocells, and the desired placement of the power lines must be specified. The result is a complete chip ready for tape-out and production, after, of course, the usual Design Rule Check (DRC) and if possible Layout Versus Schematic (LVS)<sup>7</sup> check. P&R tools are also able to route the clock network, which is a very special net: in fact, to avoid race conditions, it is important for the clock signal to arrive with almost the same delay to every device connected. A P&R tool can build a clock tree with the minimum skew.

### *Simulation*

An HDL code can always be simulated, with a specific tool, in order to test its functionality. This involves stimulus preparation, that is composed by one or more HDL modules, usually behavioral, which generate the input signal sequences for the module under test.

Simulation is crucial at every stage of the design, especially at the early ones: synthesized designs' functionality can in fact be checked only through this method. Simulating a thousands-gates physical description can be, on the other hand, extremely time consuming. The precision of the HDL description of the design increases at each stage of the design: after the synthesis the gates' delay knowledge is available, and after the place & route the HDL can be enriched with the interconnection delays. This last operation is called *back-annotation*.

It follows that simulation gets more and more accurate in the later stages, and can highlight mistakes or problems not evident at the beginning. An HDL designer will spend half of his time simulating his code<sup>8</sup>.

### *Static timing analysis*

An important check that has to be done on synchronous digital circuits is static timing analysis. In fact, synchronous circuits make use lots of flip-flops and registers to store data in pipelines and state machines. Those devices have two special timing requirements: setup time and hold time, and both have to be respected for correct circuit operation. These two timing constraints are given always in respect to the clock. Long pipeline stages tend to violate the setup time of the following registers when their delay gets close to the clock period<sup>9</sup>. Short pipeline stages instead can violate hold times when the clock is skewed: this generates a race condition.

Static timing analysis checks every pipeline for timing violations by adding up the components' timing characteristics.

---

<sup>7</sup>Meaningful LVS checks can't actually be performed on a synthesized design, because a schematic is not available since the beginning. Anyway an LVS can be done between the final layout and the physical description obtained from the synthesis.

<sup>8</sup>and the rest trying to explain himself why he had chosen to be an HDL designer.

<sup>9</sup>For a single-phase design.

### 3.2.3 Design flow

For the design of the Kchip a complete Verilog HDL description has been prepared and simulated. Then it has been synthesized, placed and routed with the tools described before. Figure 3.5 shows the design methodology followed.

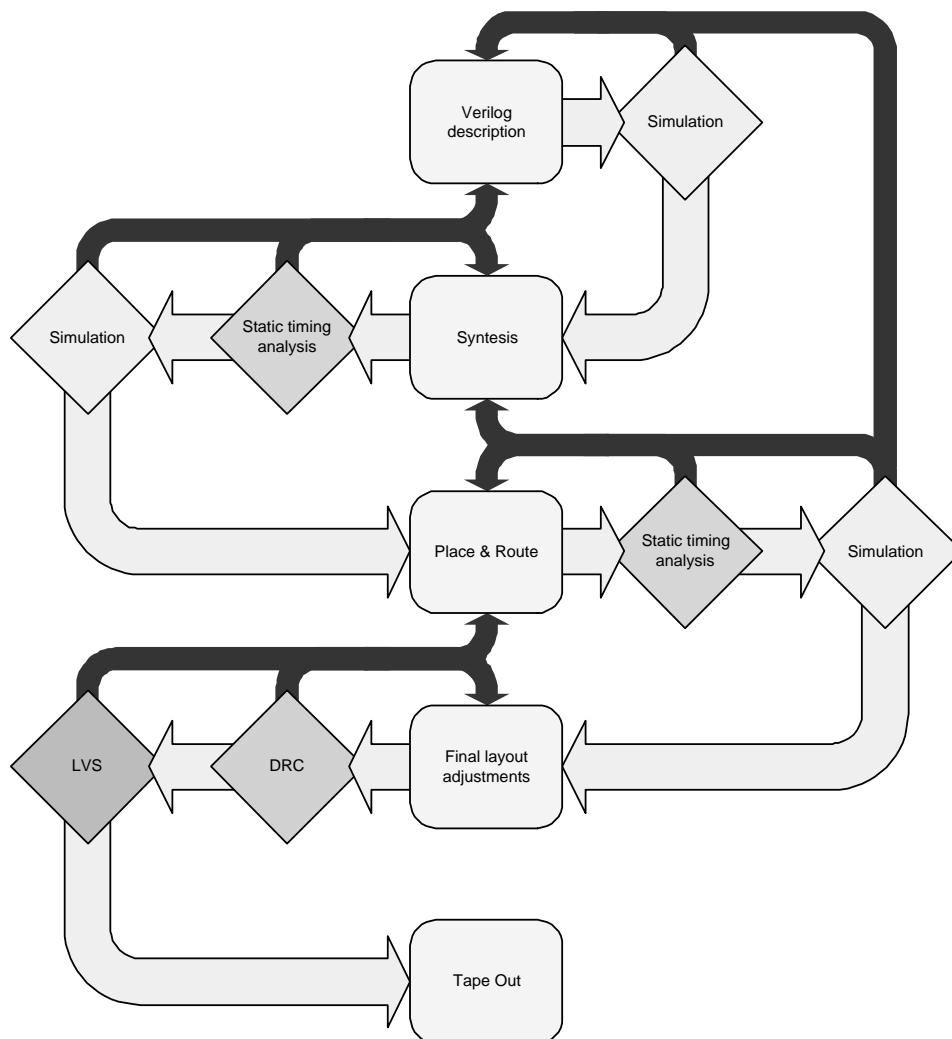


Figure 3.5: Design flow diagram.

The Kchip has been developed using the CERN's 0.25  $\mu\text{m}$  radiation tolerant library static logic components, connected in a single-phase clock configuration.

### 3.2.4 Top-level block diagram

The Kchip is divided into several internal blocks which take care of different tasks, Figure 3.6 shows the top-level Kchip block diagram.

The structure is, as usual, composed by the datapath logic along with some control logic. The blocks in the datapath are the DeDDR, the Column Addresses FIFO, the Data FIFO, the Packet Formatter, the GOL Interface and the Trigger Handler. This last block carries out also some important control jobs, as one would guess by its name, generating part of the header of the transmitted data packets. The rest of the chip, namely the PACE Controller, the Trigger Decoder, the CalPulse Builder, the Clock and Control block, the Error Logger and the I2C Block, form the control logic.

#### *Main operation*

Briefly, the T1 Level-1 trigger signal arrives directly to the Trigger Decoder from the outside CCU, and there the four fast-timing control signals are generated. These are distributed to the rest of the chip and, passing through the Clock and Control block, they reach the four attached PACEs.

Before going to the PACEs, the the LV1 trigger readout request is filtered by the Trigger Handler, which checks if there is enough space in the Trigger FIFO and also in the PACE FIFO. If this is false, no readout request will be sent to the PACE and the header of the event will contain error flags set.

The PACE Controller emulates the behavior of a PACE, building a copy of its output signals. Knowing the exact PACE sequencer's state, the PACE Controller de-serializes correctly the column addresses from the ColAddr line, and tells when the data in the ADCs output buses has to be sampled.

When the all the Kchip FIFOs contain enough data, the Packet Formatter is alerted and tries to begin its operation. This depends on the availability of the link form the GOL. The GOL Interface eventually adds a 16-bit CRC to the packet while it is sent out.

### 3.2.5 Buffers size

As resumed in the last paragraph, in order to complete its buffering functions, the Kchip needs a number of 3 FIFOs:

- the Data FIFO containing the data coming from the ADCs;
- the Column Addresses FIFO which stores the column addresses from the PACE; and
- the Trigger FIFO for the event header information which is in fact the first to be stored. This FIFO lays logically inside the Trigger Handler.

These FIFOs have to be sized to minimize the data loss probability. The choice is based on queueing theory issues: every LV1 pulse is seen as a client arrival since it requests the readout of an event, and each client is served by

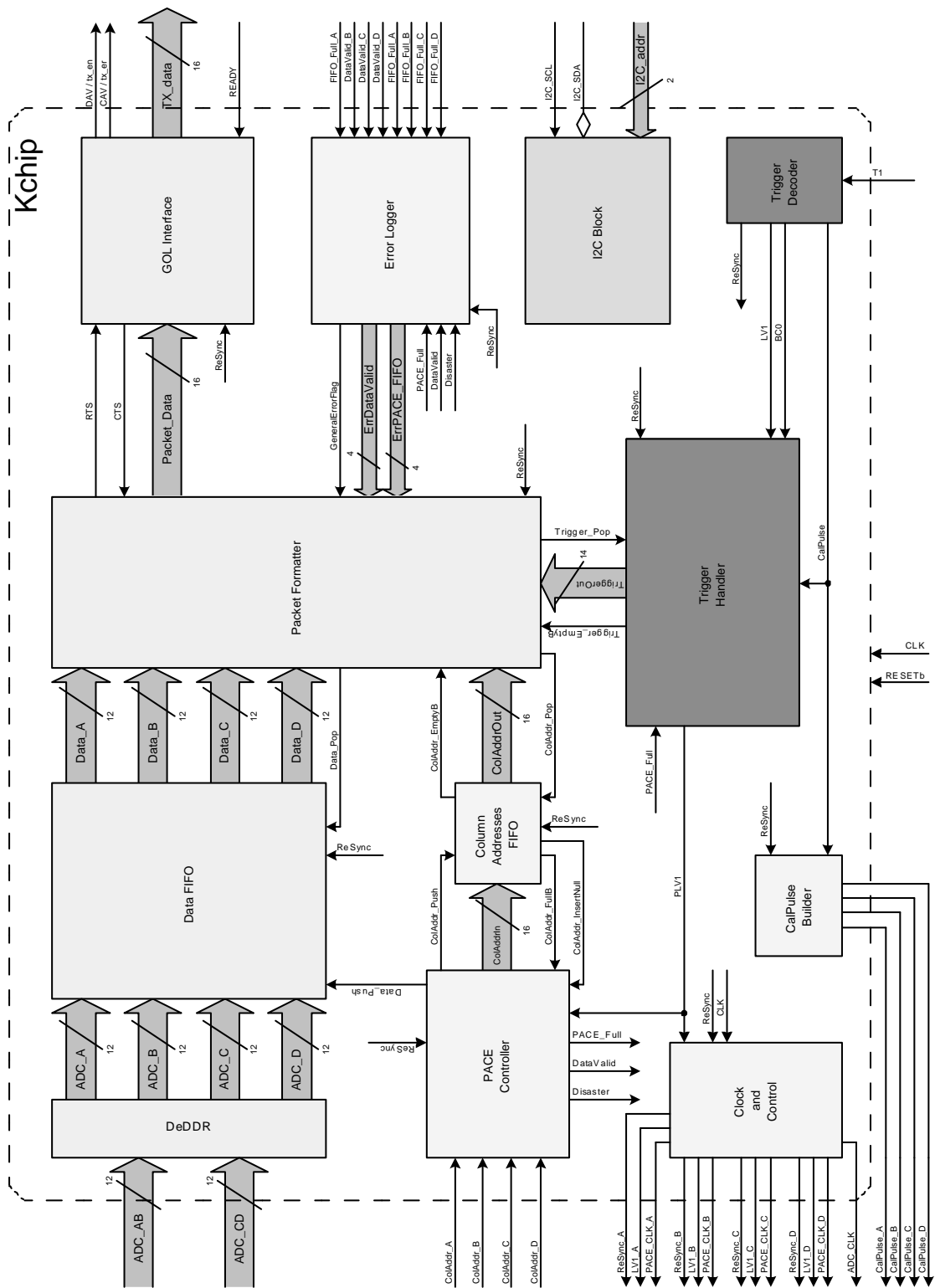


Figure 3.6: Kchip top-level block diagram.

the link. The LV1 inter-arrival mean time is modelled as a Markovian chain. From the queueing theory point of view, the Data FIFO and the Column Addresses FIFO behave in the same way, because they are fed and served at the same rates, while the Trigger FIFO doesn't. It follows that there are actually three queues in the front-end system: one into the PACE, and two into the Kchip, Figure 3.7 resumes the structure.

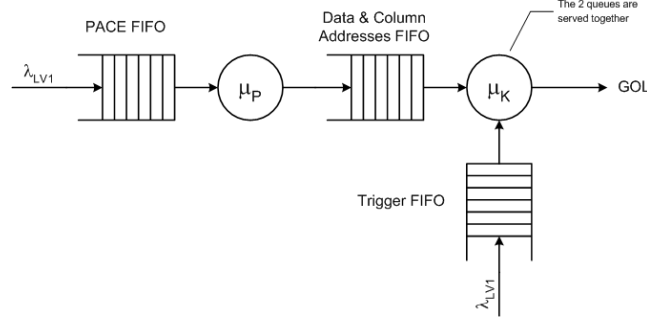


Figure 3.7: FE system queue structure. Values are  $\mu_P = 1/x_P = 145$  kHz,  $\mu_K = 1/x_K = 129$  kHz,  $\lambda_{LV1} = 100$  kHz.

The Kchip service time is defined by the packet length and, as will be explained later, is equal to  $x_K = 7.745 \mu s$ . The PACE service time instead is shorter:  $x_P = 6.9 \mu s$ . From this difference follows the absolute need for buffering inside the Kchip.

The PACE FIFO can be modelled correctly as a M/D/1/d queue and studies on its behavior were done and proved the rejection probability to be  $\approx 2 \cdot 10^{-6}$  with the given memory size.

The Kchip FIFOs don't have a simple reference model. The arrivals in the Data FIFO are not Poissonian because the PACE buffering smooths the probability density function. Since an event can be sent to the link only when all the data is ready, the Trigger FIFO clients have to wait for the other two FIFOs to be filled before being served: its service time has therefore a very complex behavior. A simplified study of these queues could be done considering again LV1 as client with its arrival rate, together with the M/M/1/d model, which is anyway a worst case estimate.

#### *M/M/1/d model*

In this case the probability of rejecting a readout request is expressed by the formula

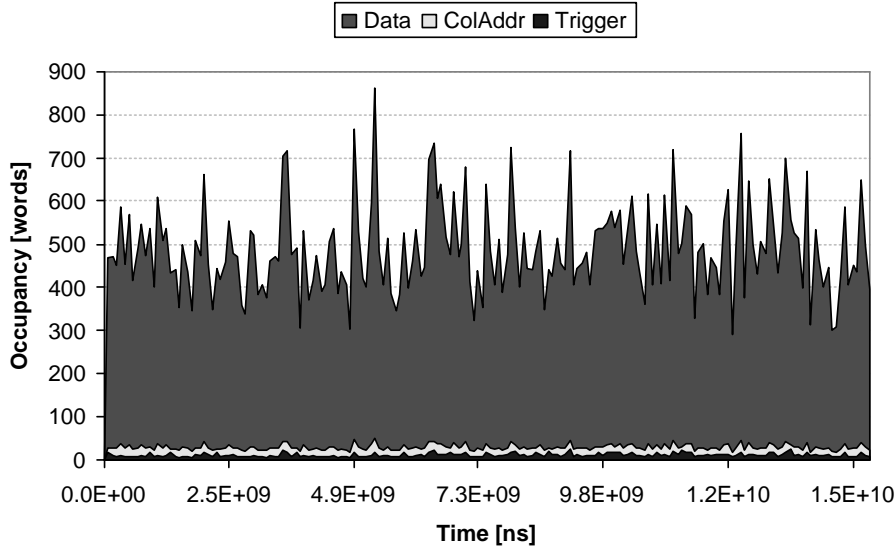
$$P_{rej} = \frac{(1 - \rho)\rho^d}{1 - \rho^{d+1}},$$

where  $\rho = \frac{\lambda}{\mu}$  is the utilization factor and  $d$  the depth of the queue.

#### *Simulations*

Anyway such a complex queue system is best dimensioned through sim-

ulation. For this purpose behavioral Verilog simulations of the front-end have been run over a large amount of events: Figure 3.8 shows one result and Table 3.4 resumes the chosen FIFO sizes.



Time examined	15.102	s
Number of events	$1.5 \cdot 10^6$	
Mean interarrival time	10.059	$\mu\text{s}$
PACE rejected events	7	
Kchip rejected events	0	
Maximum Trigger FIFO occupancy	26	words
Maximum Column FIFO occupancy	52	words
Maximum Data FIFO occupancy	863	words
Average Trigger FIFO occupancy	3	words
Average Column FIFO occupancy	2	words
Average Data FIFO occupancy	36	words

Figure 3.8: Queue system simulation result showing the FIFO occupancy over  $1.51 \cdot 10^6$  events. Maxima values are plotted in the graph, no overflows in the Kchip were observed.

As can be seen from the graph, the probability of a rejection is less than  $10^{-6}$  since more than one million events were run. The reason for this result is the PACE FIFO smaller size in respect to the Kchip FIFOs: the PACE rejection ratio is therefore bigger and filters out the arrival rate to

FIFO	Depth [events]	Depth [words]	Implementation
Data	$10 + \frac{2}{3}$	1024	$1024 \times 18$ bit SRAM ( $\times 4$ )
Column Addresses	$10 + \frac{2}{3}$	64	$128 \times 27$ bit SRAM
Trigger	64	128	$128 \times 27$ bit SRAM

Table 3.4: Kchip FIFOs size.

the Kchip.

### 3.2.6 The Clock and Control block and synchronization goals

Since the PACE, the ADC and the Kchip are cascaded, they have to respect mutually their timing requirements. Unfortunately none of these chips is arrived at the final stage of engineering yet, thus their complete characterization is not available: the references are therefore the design goals which can have fluctuations at the production stage. The synchronization among the chips has to be set up with particular care, keeping in mind these issues and making the system work in worst case conditions.

#### ADC requirements

The ADC requires the analog input to be stable in a 2 ns interval centered on the clock rising edge. The PACE settling time is about 35 ns, but this value doesn't violate any constraint: since the output period is 50 ns the ADC requirement is met by far. On the other hand the PACE output becomes floating very rapidly after the clock rising edge. During the Kchip design we considered the PACE floating time as zero. It follows that, if the PACE and the ADC are driven by the same clock, there will be a 1 ns violation on the ADC's hold time. Figure 3.9 shows the PACE-ADC timing diagram with violation.

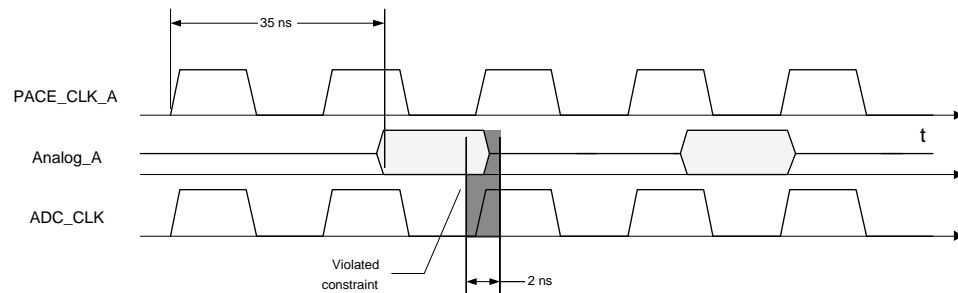


Figure 3.9: PACE-ADC timing diagram showing possible ADC requirement violations.

The adopted solution is to delay the PACE clock in respect to the ADC



clock by more than 1 ns, using an inverter chain. In this way also the PACE output will be delayed, since is relative to the clock. Figure 3.10 shows this solution<sup>10</sup>. A simple chain of 24 inverters guarantees a typical delay of 1.3 ns.

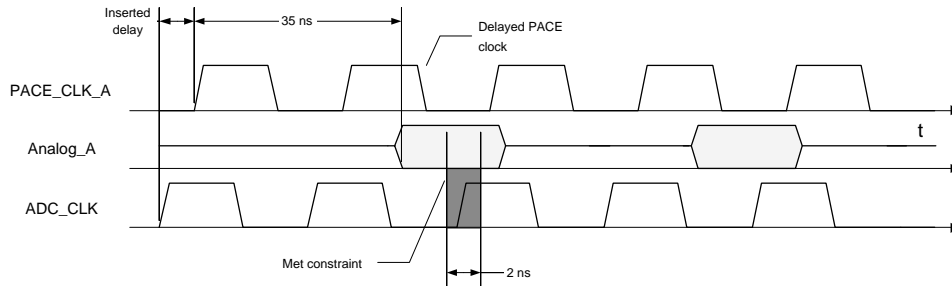


Figure 3.10: Solution fixing the violated ADC requirements.

The Kchip itself has also a few requirements in respect to the ADC: since the ADC output is transmitted with the Double Data Rate (DDR) technique, the Kchip internal registers timing constraints have to be respected on both the clock edges. Because of its internal operation, the ADC has a long “positive clock edge to data” characteristic but a short “negative clock edge to data”. Moreover, the internal Kchip clock tree has a non-zero delay (as will be seen later, nearly 1 ns) from the root input pin to the registers, which have to store the ADC data.

*Kchip requirements*

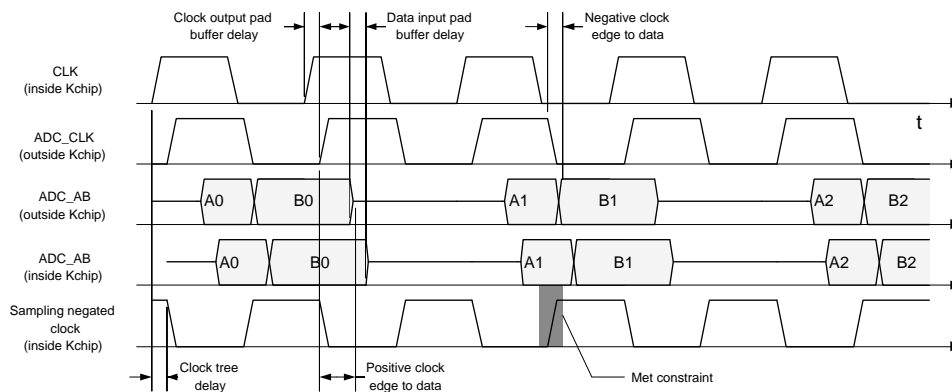


Figure 3.11: ADC-Kchip timing diagram showing the Kchip internal registers’ requirements.

<sup>10</sup>One another solution could be giving a negated clock to the ADC in respect to the PACE clock.

To avoid problems on the hold time of the negative edge sampling registers inside the Kchip, the ADC clock is generated internally the Kchip and propagated outside: the clock output pad buffer, together with the data input pads' buffers, give a sufficient delay. Figure 3.11 clarifies the interaction between the ADC and the Kchip.

### *PACE requirements*

Eventually, in order to respect the PACE digital sequencer timing requirements, the filtered Level-1 trigger readout request PLV1 and the ReSync signal are delayed by half clock cycle using negative edge flip-flops. The criticality of those signals is high, thus the flip-flops are triplicated to avoid pulses due to SEUs.

### *Block schematic*

Figure 3.12 resumes the Clock and Control block schematic.

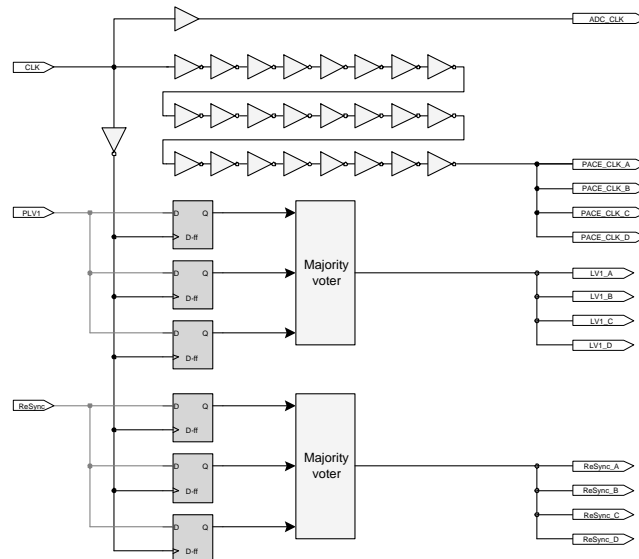


Figure 3.12: The Clock and Control block's schematic.

### 3.2.7 The Trigger Decoder

As mentioned before, the Trigger Decoder takes care of decoding the four fast-timing control signals from the Level-1 trigger T1. This is done through a simple state machine which looks for '1's in the T1 stream and then counts up to 3 clock cycles before allowing the decode of the command. Figure 3.13 illustrates the Trigger Decoder state machine diagram.

The T1 stream is stored in a 3-bit shift register and every time the state machine is in the DECODE state a combinatorial logic part calculates

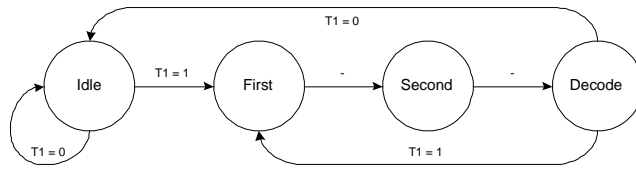


Figure 3.13: The Trigger Decoder’s state machine diagram.

the outputs. These are also stored in a 4-bit register that keeps the last command issued (LAST\_T1CMD), accessible via I<sup>2</sup>C for testing purposes.

The possibility to mask commands is available: another 4-bit register, MASK\_T1CMD situated in the I2C Block, enables or disables each one of the four fast-timing control signals: when one of its bits is set, a command is disabled. Moreover a general enable signal is present: for testing reason, when KchipMode is asserted all the commands are masked. Figure 3.14 shows the Trigger Decoder schematic.

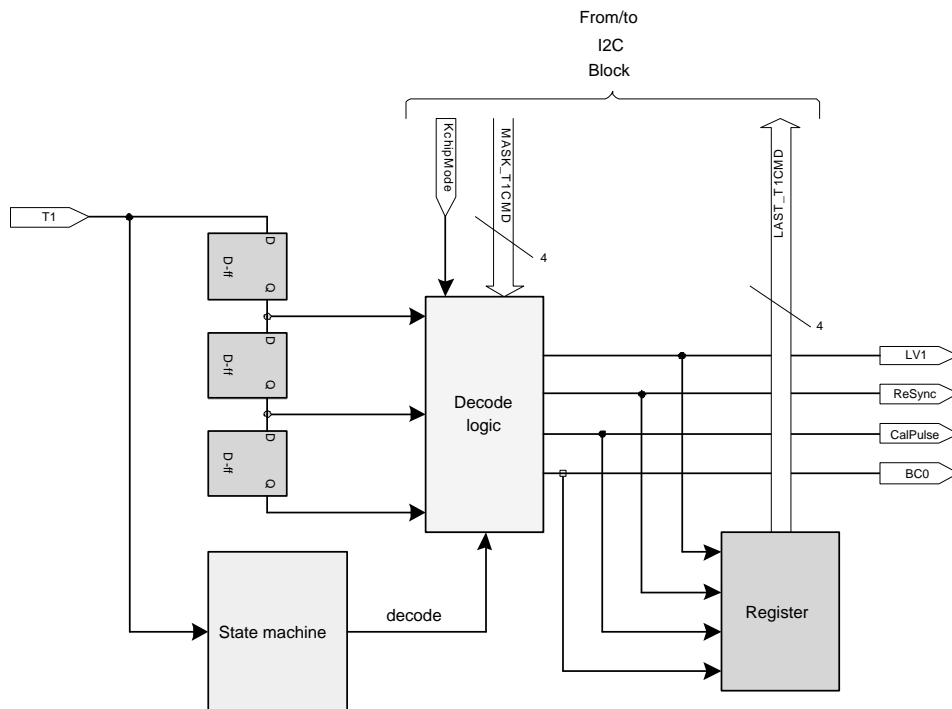


Figure 3.14: Trigger Decoder schematic.

### 3.2.8 The Trigger Handler

From the Trigger Decoder the LV1 readout request goes directly to the Trigger Handler. This last block is composed by two main subunits: the Trigger FIFO and the Trigger Controller. Together they prepare and store the first header data for the event, which is composed mostly by counters and flags describing the status of the system at the arrival instant of the LV1 pulse. Figure 3.17 shows the Trigger Handler schematic.

#### Trigger Controller

The Trigger Controller is in practice a state machine plus some counters. The counters are needed for event order reconstruction in the counting room: the event builder has to find references in each received packet to assign it to the right bunch crossing. There are essentially two counters in the header:

- the 12-bit **BunchCounter** which is incremented at every clock cycle, representing a specific bunch crossing; and
- the 8-bit **EventCounter** which is incremented at every LV1 pulse.

Both the two counters are reset on a **ReSync** pulse and on a **BC0** pulse. This last command in particular is implemented just for this function. The **BunchCounter** is also sampled on every LV1 arrival and kept by a register accessible via I<sup>2</sup>C called **LAST\_BC**.

For each arriving LV1 pulse, the logic checks two input signals indicating the availability of space in the Trigger FIFO and in the PACE FIFO, called respectively **Trigger\_Full** and **PACE\_full**. While these signals are low, the PLV1 output is an exact copy of the LV1 input. Instead, if one of the above mentioned FIFOs gets full, the PLV1 will stay low, preserving the PACE from receiving a readout request. This trigger inhibition functionality can be turned off by raising the **TriggerInhibitMode** input coming from the I2C Block: in that case the PLV1 will never be suppressed.

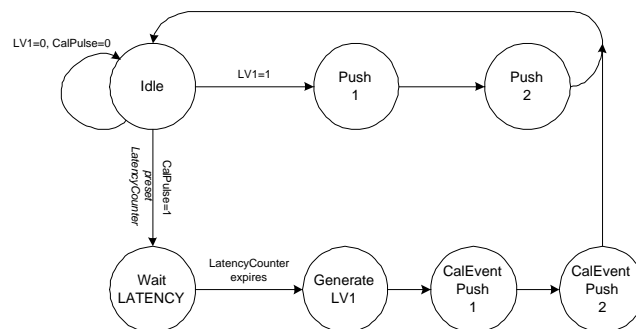


Figure 3.15: The Trigger Controller's state machine diagram in the default configuration.

The Trigger Controller state machine reacts to the LV1 and CalPulse signals in order to push the correct flags into the Trigger FIFO. Moreover, it generates internally a LV1 pulse for the PACE whenever a calibration pulse is sent: this is called a calibration event and the block can be configured to mask it through the MaskCalLV1 input. Since the calibration pulse serves the PACE pipeline analog memory, the readout has to be performed 128 clock cycles later: a third 8-bit counter, LatencyCounter, preset to the LATENCY I<sup>2</sup>C configurable value (128 by default), starts counting down to zero when the CalPulse is received. The LatencyCounter is decremented at every clock cycle and only when it has expired the virtual LV1 pulse is issued. Figure 3.15 describes the Trigger Controller state machine.

As can be seen from the diagram, the state machine departs from the IDLE state in two directions: one in case of a LV1 pulse and another in case of a CalPulse. Both the paths have two PUSH states: the Trigger Controller pushes, in fact, two 14-bit words into the Trigger FIFO per event. The first word contains the BunchCounter while the second the EventCounter. The remaining bits are used by flags: Figure 3.16 shows the data structures pushed.

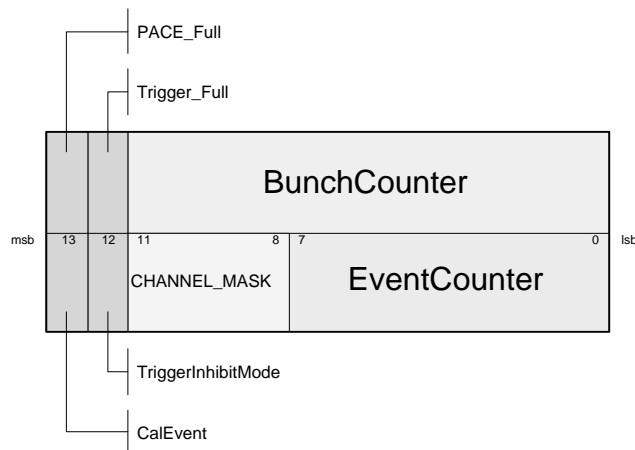


Figure 3.16: Data structure stored into the Trigger FIFO for each event.

Together with the EventCounter, a 4-bit field describing the status of the four PACE channels is packed: this is the CHANNEL\_MASK. A ‘1’ in it means that the corresponding channel is active, while a ‘0’ turns it off. In fact, the four PACE could in principle be out of synchronization or malfunctioning, thus it follows the need of signaling which data should be considered as valid and which not.

Four flags are stored along with the counters: three of them are input signals of the Trigger Controller, while the fourth, the CalEvent bit, is generated internally and is true when the state machine is in one of the two

CALEVENT PUSH states. Therefore only calibration pulse readout events will have the CalEvent flag set.

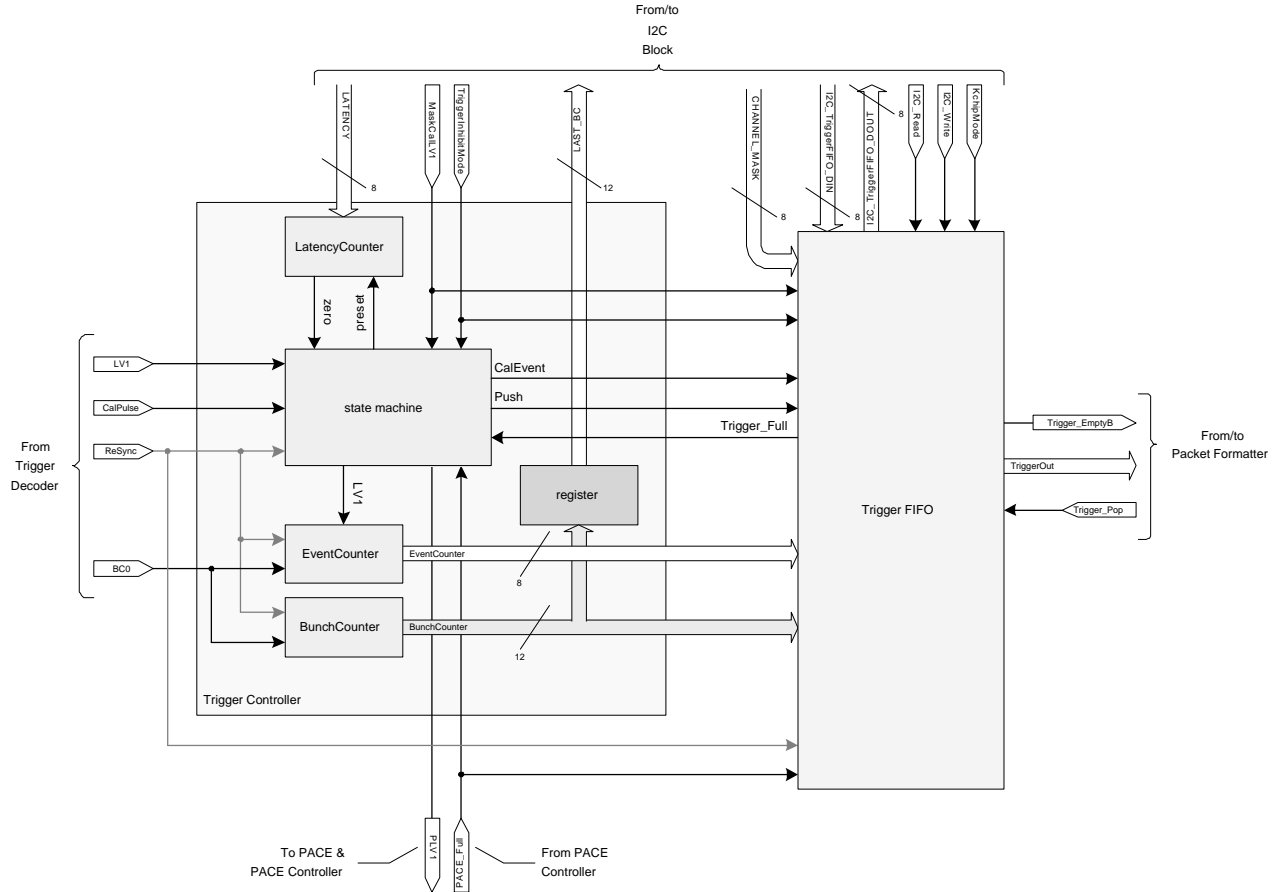


Figure 3.17: Trigger Handler schematic.

### Trigger FIFO

As described in Table 3.4, the Trigger FIFO exploits a 128-words deep, 27-bit wide, dual-port SRAM<sup>11</sup>, which is available in the cell library [20], plus a FIFO controller. The controller is again a state machine.

The choice of 27-bit wide FIFO memory, even though the data is composed by 14-bit words, is mainly due to the availability in the library, nevertheless it is possible to use the remaining bits for forward error correction redundancy. In the present version of the Kchip, almost no redundancy is implemented, while is foreseen the presence of coding logic in the future versions.

<sup>11</sup>The memory is actually made of single-port cells but it uses the positive edge of the clock cycle for reading and the negative edge for writing, thus it is seen as dual-ported from the outside.

The only data to be protected with redundancy is the flags: the two flag bits laying in each word are triplicated before being written in the memory<sup>12</sup>. A majority voter restores the original data structure in the memory output. The reason for this comes from the fact that the two `PACE_Full` and `Trigger_Full` flags are crucial for the behavior of the system: they in fact both signal the impossibility of reading out the corresponding event.

The `Trigger FIFO`, as well as the other FIFOs in the Kchip, has the peculiarity of being able to be accessed via I<sup>2</sup>C interface: for versatile memory testing purposes the FIFO can be easily written and read directly when the `KchipMode` signal is set to true. Since the I<sup>2</sup>C communication protocol provides transmission of a byte per transaction, each FIFO operation has to be completed in two I<sup>2</sup>C accesses. This also implies that, during a write operation, the data can be pushed only when the second I<sup>2</sup>C transaction is performed, and a register is needed to store the first part of the information. Viceversa, during a read operation, the data has to be popped only when the first I<sup>2</sup>C transaction takes place. Therefore, to implement this functionality, the data will be pushed only when writing to the least significant byte (LSB) and will be popped only when reading from the most significant byte (MSB).

One last consideration about the `Trigger FIFO` is that the high-active `Trigger_Full` signal is true not only when the queue is actually full, but also when there is space left for just one pair of words. In fact, when the FIFO gets full, the `Trigger Controller` has to communicate the impossibility of receiving more readout requests, and this can be done only if there is enough space in the FIFO to push the data structure described above. A similar rule is provided for the low-active `Trigger_EmptyB` signal, which goes to the `Packet Formatter`: only when at least two words are present in the FIFO the signal is true, since the packet formatter will pop two words per event.

### 3.2.9 The PACE Controller

The `PACE Controller` emulates the PACEs behavior from an external point of view: in fact, it

- calculates the status of the `PACE FIFO`, generating the `PACE_Full` signal;
- sequences a readout operation, building the `DataValid` signal;
- knowing the ADC's number of pipeline stages, sends the `Data_Push` command to the `Data FIFO`;

---

<sup>12</sup>The two triplicated bits are also interleaved one with the other in order to minimize the possibility of double SEU hitting two out of three locations. In fact, tests on the library SRAMs proved that double SEU on adjacent memory cells can occur with a non-negligible probability, while triple SEU are by far more unlikely.

- de-serializes the four PACE column addresses lines and puts the result into the Column Addresses FIFO.

A lot of functionality is therefore implemented in this block, Figure 3.18 illustrates its schematic.

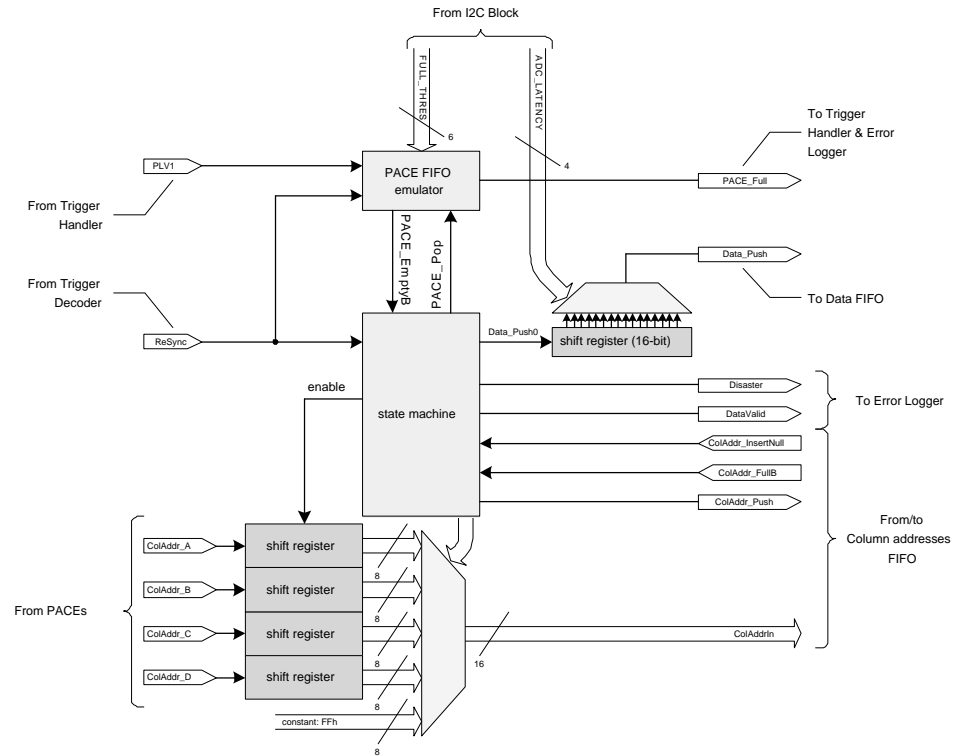


Figure 3.18: PACE Controller schematic.

The need for a Kchip internal copy of the signal coming from the PACEs follows from the fact that these chips can be out of synchronization one from the other and, on top of that, a variable number of PACEs, from 1 to 4, can be connected to the Kchip, preserving the same functionality. Thus, a voting decision among the four `FIFO_full_x`, as well as among the `DataValid_x` signals can't take place: in general, it is impossible to vote among an even number of signals, since a conflicting fifty-versus-fifty situation can happen, leaving the signal undecided<sup>13</sup>. The PACE Controller emulated signals are therefore used in place of the incoming ones.

#### *PACE FIFO emulator*

In order to produce the `PACE_Full` signal, a simple 6-bit counter is used: when the counter is above a fixed threshold, the signal becomes true. The

<sup>13</sup>unless if a “golden” signal is provided.



counter is incremented 3 times at every incoming PLV1 and decremented once every time a PACE column is read. The threshold is set by an I<sup>2</sup>C accessible register called `FULL_THRES`, with a default value of 42 (while the PACE FIFO depth is 45 words). Lowering this value will make the PACE FIFO to be filled less than its actual size. Raising the threshold, instead, would possibly overflow the FIFO, with unrecoverable losses of data.

The four column addresses are de-serialized by a set of shift registers and then packed in a structure formed by two consecutive 16-bit words, represented in Figure 3.19, by a multiplexer. The shift registers are enabled only during the right time interval.

*Column addresses  
de-serialization*

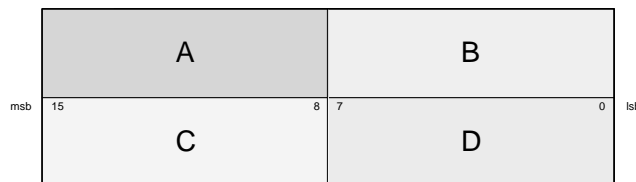


Figure 3.19: Data structure pushed into the Column Addresses FIFO for each column read. The letters indicate the source of the values.

Again, an emulation of the PACE sequencer has to be done to generate a `DataValid` signal. A state machine takes care of this emulation and follows exactly the timing diagram already described in Figure 3.4.

*Readout sequencer*

Moreover, the state machine controls the Data FIFO and the Column Addresses FIFO by receiving the `ColAddr_FullB` and by sending them the push command. A FIFO full flag signaling from the Data FIFO is not needed since it has exactly the same depth, in terms of events, of the Column Addresses FIFO, thus only this last block is checked. The two FIFOs will get filled together and read together.

When the `DataValid` becomes true, the column addresses are sent through the serial line and the shift registers are enabled. Then, after 8 clock cycles, the output of the shift registers is ready. They are therefore frozen and the `ColAddr_Push` is pulsed to fill the corresponding FIFO. The pulse lasts 2 clock cycles, since two words have to be written. Figure 3.20 shows the state machine's timing diagram. Since 3 columns are read per event, 6 words will be stored in the Column Addresses FIFO.

The `ColAddr_InsertNull` signal notifies the PACE Controller that the Data FIFO is full: it becomes true when 10 events are written in the Column Addresses FIFO. If this happens when the PACE is sending data, an error flag data structure of the same size is pushed in the FIFO instead of the actual column addresses, since there won't be enough space for the data.

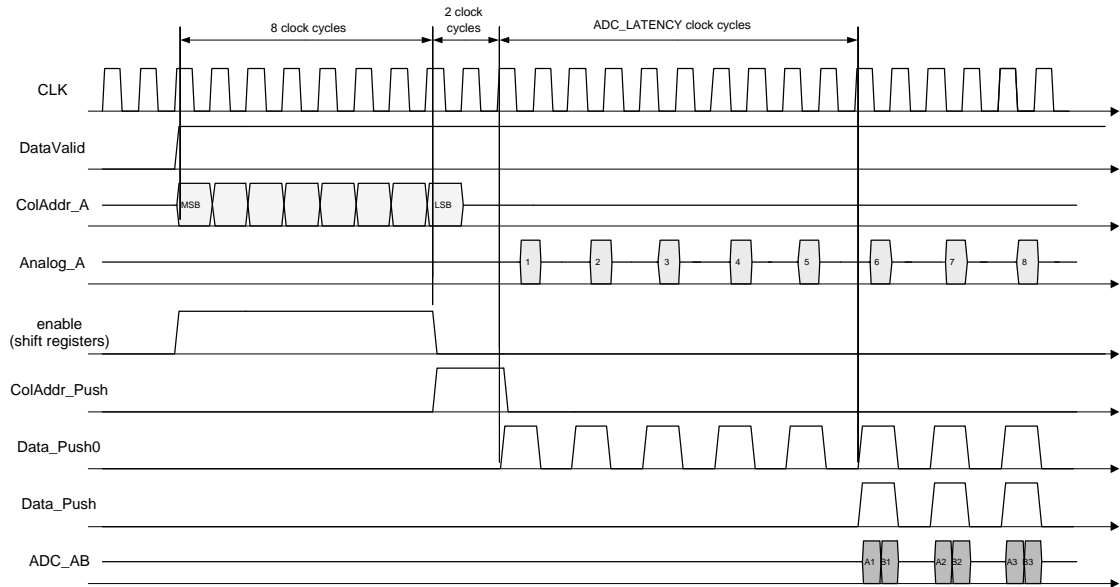


Figure 3.20: PACE Controller state machine's timing diagram.

The structure is composed by all '1's: no real column address will have a conflicting value since the valid range for them is 0 – 160.

In very pessimistic cases, in practice only when the optical link is inactive, the Column Addresses FIFO could actually overflow. Even if it is very unlikely, this condition is especially dangerous for the system: the Kchip would contain in fact partial events in its FIFOs, since some information would be already stored in the Trigger FIFO, and will not be able to resume a correct operation until a re-synchronization command is issued. Thus, when the Column Addresses FIFO overflows a special error signal is raised: the *Disaster* flag. This flag will be read by the counting room as soon as the link returns in a working condition and will be reset only when a *ReSync* pulse arrives.

The *Data\_Push* command is built from an intermediate signal called *Data\_Push0*. This last signal is strobed each time the PACEs are normally putting valid analog data on their output lines: by delaying *Data\_Push0* as many clock cycles as the number of ADC pipeline stages, the *Data\_Push* is obtained. This delay is performed by another 16-bit shift register. A multiplexer selects among the 16 stages of the shift register: in this way, the delay is programmable via I<sup>2</sup>C setting the value of the register *ADC\_LATENCY*, which drives the multiplexer select lines. The default value of *ADC\_LATENCY* is 5, equivalent to the AD41240's number of stages.

As mentioned before, the *Data\_Push* signal will pulse 32 times per column read, therefore 96 data words will be filled into the Data FIFO per event.

### 3.2.10 The Error Logger

The main tasks of the Error Logger are to check the PACEs synchronization, count the number of SEU occurrences in the Kchip's registers and report the errors to the Packet Formatter. Figure 3.21 describes the block's schematic.

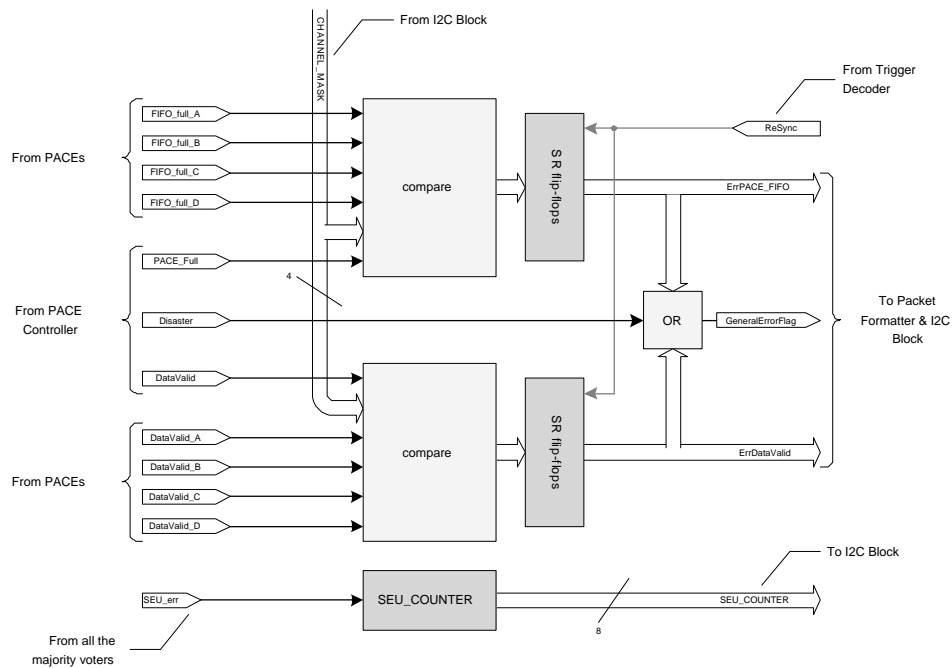


Figure 3.21: Error Logger's schematic.

The two signals emulated by the PACE Controller are compared, inside the Error Logger, with the corresponding PACE outputs. A combinatorial logic block receives the four `FIFO_full_x` and the four `DataValid_x` signals and cross checks them to the internal `PACE_Full` and `DataValid` values respectively. In practice the internal Kchip values are considered “golden”, and the PACEs must follow these signals. The inactive channels are masked by the `CHANNEL_MASK` bus, therefore they're not compared and no error is reported in case of differences. The results are registered by a set of SR flip-flops, namely `ErrPACE_FIFO` and `ErrDataValid`: an error will set true the respective register content, but only a `ReSync` pulse will be able to clear the value. The registers are accessible via I<sup>2</sup>C.

*PACE  
synchronization*

Even if not mentioned before, all the Kchip's majority voters have an error output, besides the voted output, which tells when the input signals differ. In this way all the triplicated state machines are checked for SEU

*SEU count*

occurrence, and all their errors are reported to the **Error Logger** through a big OR-gates tree. In other words, all the state machines' error outputs are ORed together to obtain a single **SEU\_err** signal. Only a SEU per clock cycle can be counted: multiple errors within a cycle will be reported as one. Nevertheless the probability of multiple SEU within a cycle is rather small.

The **SEU\_err** signal increments an 8-bit counter called **SEU\_COUNTER** which is accessible via I<sup>2</sup>C interface in order to keep SEU occurrence statistics. When the **SEU\_COUNTER** reaches its maximum value (that is 255), it stops and remains in that value until a hard reset is issued (**RESETb** low). The counter is not reset on a **ReSync** pulse.

The **SEU\_COUNTER** represents also a self-test circuit for the chip: if faults are present in one of the state machines after the fabrication, the counter is very likely to be incremented, since the error wouldn't be triplicated and would give rise to errors.

### *Error reporting*

A special flag is reserved in the link packet for errors: this is the **GeneralErrorFlag**, which is generated inside the **Error Logger** ORing bitwisely the two **ErrPACE\_FIFO** and **ErrDataValid** buses plus the **Disaster** bit coming from the **PACE Controller**. The resulting flag reports general loss of synchronization in the front end system, and it is sent to the **Packet Formatter**, which includes it in the link data stream. When the **GeneralErrorFlag** raises, the front end systems requires an immediate resynchronization.

#### **3.2.11 The DeDDR**

The **DeDDR** is simply composed by a set of negative edge flip-flops which sample the data input from the ADCs, extracting the **B\_ADC** and **D\_ADC** data channels. The other two channels are just feeded through the block.

#### **3.2.12 The Data FIFO**

The chosen implementation of the **Data FIFO** uses 4 identical 1-Kword deep, 18-bit wide SRAMs. The four memories are filled respectively with the data coming from the four PACEs, but, at the same time, they are connected to the same FIFO controller. In practice, they behave like a single wide SRAM.

Again, since each channel uses only 12 bits of the 18 available, in future developments the remaining bits can be used for forward error correction.

Through the **CHANNEL\_MASK** input bus, it is possible to deactivate part (or all) of the data channels, filling the corresponding memory with zeros. As described before, the **CHANNEL\_MASK** register lays in the **I2C Block**.

It is also possible to test the FIFO via I<sup>2</sup>C interface: since it is a very wide memory a special technique has to be used to perform the access through the 8 bit bus. However, the test patterns employed will often be the same

for all the 4 SRAMs, thus an efficient way to fill them is to write them contemporarily with the same input, saving precious time. Two ways to store data in the FIFO through I<sup>2</sup>C are therefore made available:

- a slow mode, filling one memory per time, writing different data from one to the other;
- a fast mode, filling all the memory together, writing the same data.

A 4-bit wide address bus, part of the I<sup>2</sup>C register called `I2C_FIFO_select`, is used to select among the four memories. The lsb of the bus selects channel A, while the msb selects channel D. Each memory can be selected independently having its own selection bit, and it is also possible to select a set of FIFOs together. Since the data has to be always written in the four SRAMs at the same time, a few registers are used to keep the information until all the 4 words are transferred. Although it is possible to select any configuration of the FIFOs when writing, the operation is thought to be performed with a specific order: from channel A to channel D, and sending always the MSB first. Only when the LSB of channel D is sent, the push command is issued.

Of course, the read operations have to be carried out transferring each word independently. The opposite rule is valid for popping data in respect to pushing: only when the MSB of channel A is read, the data is popped. Therefore the read order is the same than when writing: from channel A to channel D, and picking always the MSB first.

I<sup>2</sup>C access to the FIFO is allowed only when the `KchipMode` flag is set true.

### 3.2.13 The Column Addresses FIFO

The Column Addresses FIFO is implemented with a 128-word deep, 27-bit wide SRAM. Only 16 of the available bits are used and the rest is reserved for future error correction coding.

On top of that, only half of the 128 locations are employed for storing real addresses: the memory is designed to contain data for as many events as the Data FIFO (that is  $10 + \frac{2}{3}$  events = 32 columns). The reason is that the Column FIFO generates the `ColAddr_InsertNull`, which is read by the PACE Controller to know when to reject data from the PACEs.

The rest of the FIFO is used to store eventual error patterns, one for each event rejected by the PACE Controller.

The SRAM is accessible via I<sup>2</sup>C in the same way as the Trigger FIFO described before.

### 3.2.14 The Packet Formatter

Following the datapath, the next block met is the Packet Formatter, which reads the data from all the Kchip FIFOs and organizes it in a packed structure suitable for transmission.

*Output packet structure*

The goal is to obtain a packet structure like the two illustrated in Figure 3.22. The second structure is used in normal operation, while the first is sent only in case of error signaling or impossibility to read the experimental data. This last situation is referred as a *null event*.

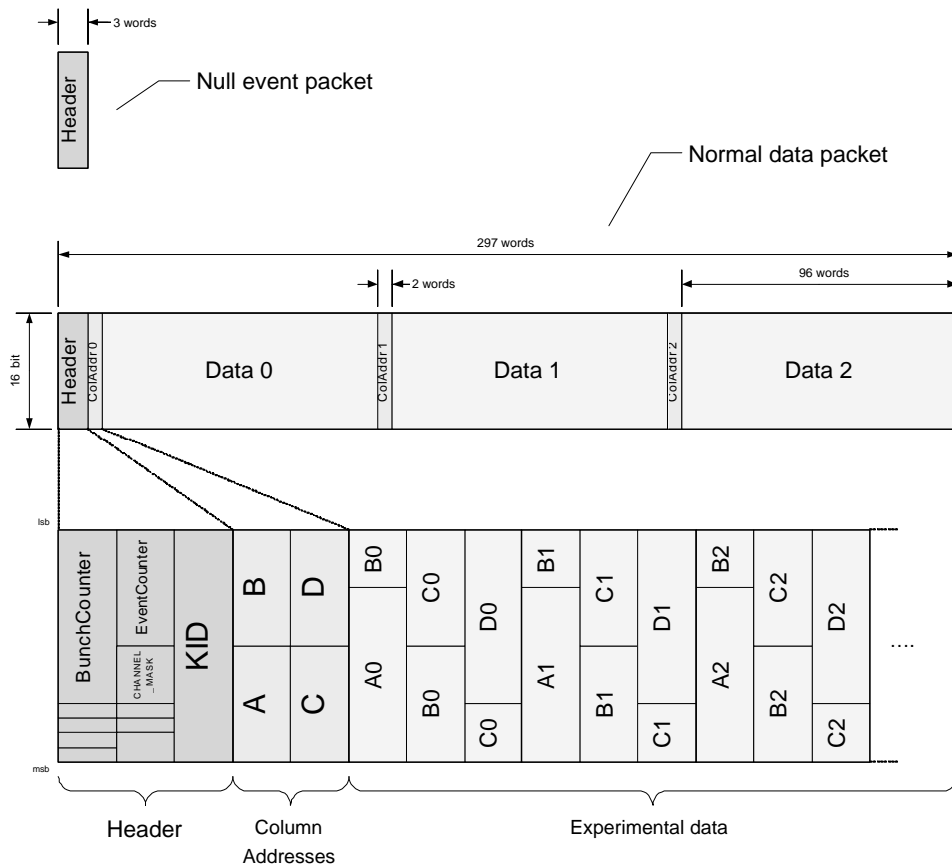


Figure 3.22: Packet Formatter output structure.

Since the Gigabit Optical Link chip accepts 16-bit data in the chosen slow mode, the packet is 16 bit wide. The packet always begins with a 3-words header, described in Figure 3.23, which contains the information gathered from the Trigger FIFO plus other three important fields. These are the 16-bit KID identifier, the Null\_Data flag and the GeneralErrorFlag.

The KID, which is partly stored into an I<sup>2</sup>C register and the rest hard-wired on the motherboard, contains a unique identifier for the chip in the whole preshower. This can be used when cabling the fibers to the counting room to check if the data comes from the correct source.

The `Null_Data` flag is calculated when analyzing the contents of the Column Addresses FIFO: as mentioned before, an error 2-words structure will be present in the FIFO when it comes close to overflow and readout can't be carried out. This structure is composed by all '1's and is recognized by the Packet Formatter, which will start a null event transmission instead of a normal one, since no meaningful data has been written into the Column Addresses and Data FIFOs. Since this condition is crucial for the success of the readout and since the occurrence of SEUs in the FIFOs can possibly alter their data, the all-'1's error pattern is requested to be in 2 out of the 4 column address fields: a special bit-wise majority voting is in fact performed on those bytes.

The `GeneralErrorFlag` comes from the Error Logger and, as described, is an alert for the counting room requesting a timely re-synchronization of the front end.

Moreover, the Packet Formatter performs an elaboration of the four `CHANNEL_MASK` flags adding error information coming from the Error Logger through the 4-bit `ErrDataValid` and `ErrPACE_FIFO` buses: in case of any error on one of the channels, the respective mask bit will be set, in order to let the counting room know which data to ignore.

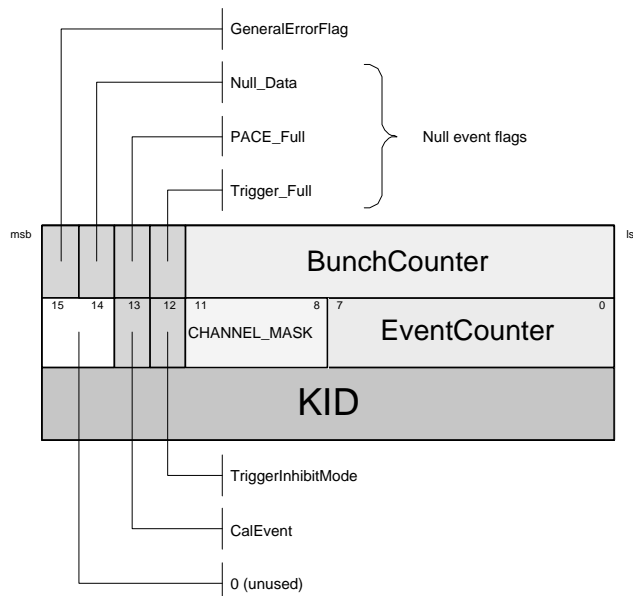


Figure 3.23: Packet header.

The triplet formed by `Null_Data`, `Trigger_Full` and `PACE_Full` resumes the null event condition: if just one among them is true, the packet sent is a null event rather than a normal one. Therefore the receiver has to look at their value to know the exact packet length, which is 297 words in case of normal event, while only 3 in case of null event. In fact, a null event is composed by only the header: from that, the counting room can still acquire important information about which events were lost.

Normally, after the header, the data stream begins: this is composed by the content of three PACE memory columns, each one preceded by its address. The content of the Column FIFO is directly put into the packet before each data segment. The  $4 \times 12$ -bit data coming from the Data FIFO is rearranged in a compact  $3 \times 16$ -bit block in order to optimize the data throughput. The block is shown in Figure 3.24. A data segment is made out of 32 blocks, thus its length is 96 words. Three segments plus their 2-word column addresses form the packet load, which is 294 words long.

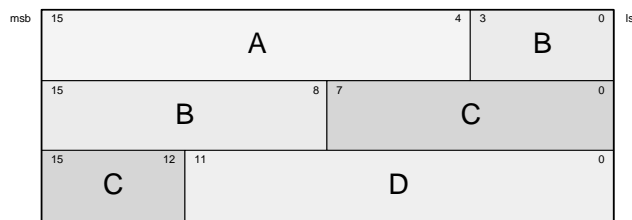


Figure 3.24: Data block.

When a data input channel is disabled through the `CHANNEL_MASK` the respective data stream will contain all zeros. This means that even if the Kchip is connected to less than 4 PACEs, the packet structure will remain the same, filling with zeros the unused slots.

### *Circuit operation*

The Packet Formatter schematic is represented in Figure 3.25. A multiplexer is used to put the right data on the output bus, while a state machine drives its select lines. The state machine is rather complex, but it follows mainly two paths: one for normal events and one for null events. Its task is also to create the pop signals for the FIFOs with the right timing, and to interact with the GOL Interface through a specific handshake. It is very important to insert the smallest number of wait states between two burst packet transmissions in order to maximize the data throughput.

Before sending any data, the state machine remains into its WAIT states until all there is enough information in the FIFOs to build a meaningful packet: this is done checking the two low-active `Trigger_EmptyB`, `ColAddr_EmptyB`, in the given order. The Trigger FIFO will in fact be the first to be filled, while the Data FIFO will be the last.



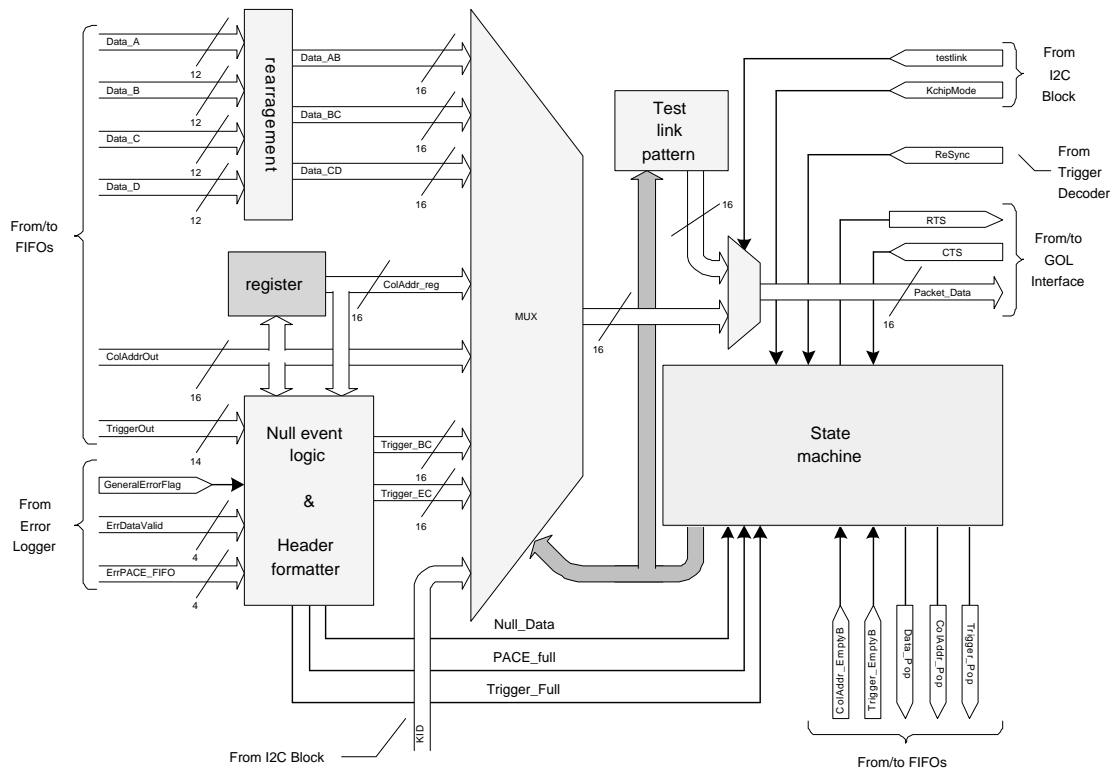


Figure 3.25: Packet Formatter schematic.

At the beginning of the transmission, the Packet Formatter has also to decide between sending a null or a normal event. Things get more complicate if one thinks that the information necessary to perform this task may come in two different moments: while the `Trigger_Full` and the `PACE_Full` flags are stored together in the `Trigger FIFO`, the `Null_Data` is available only when the `Column Address FIFO` is ready. Moreover, if just one of the first two flags is true, no data has to be extracted from the `Column Address` and `Data FIFOs` and the third flag shouldn't be checked. Figure 3.26 illustrates a simplified state machine diagram, where the beginning of the transmission is highlighted. As can be seen, the `Null_Data` flag is not checked until when transmitting the first header word (in the `SEND BC` state).

The communication signals with the `GOL Interface` are two: the `GOL interface` receives the `RTS` signal and answers with the `CTS` message. In general, the `Packet Formatter` acts as a master in respect to the `GOL Interface` which responds with a one-cycle latency to the commands. In practice, when the `RTS` is true, the `Packet Formatter` is ready to send data, and the `GOL Interface`, if the link is ready, answers after one clock cycle asserting the `CTS`. The `Packet Formatter` receives the clearance and after another clock cycle the data is sent. After a transmission is initiated, the `RTS` will not

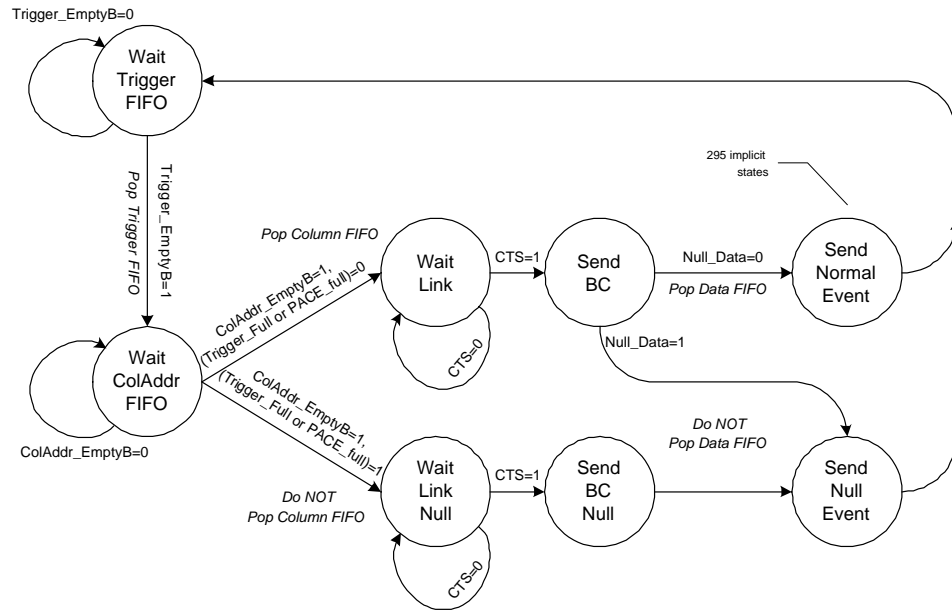


Figure 3.26: Simplified Packet Formatter state diagram.

be deasserted until the whole packet is sent, and this will happen even if the CTS lowers in the middle of the operation. In this case the sent data will be irreparably lost. To signal the end of the packet transmission the RTS will become false for at least one clock cycle, pulsing in case of burst transmissions. Figure 3.27 describes the handshake.

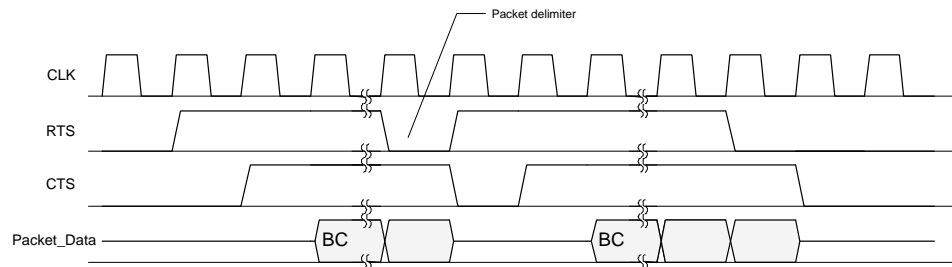


Figure 3.27: The Packet Formatter-GOL Interface handshake's timing diagram. A 2-packets burst is shown.

The `testlink` input, driven by an I<sup>2</sup>C register, when true, switches a second multiplexer present in the datapath, connecting the output bus to a link test pattern generator. The pattern is fixed and has been chosen to be easy to detect by the counting room.

Eventually, the `KchipMode` bit disables the whole block and forces the

state machine to remain into the WAIT TRIGGER state.

### 3.2.15 The GOL Interface

The GOL Interface is the last block in the datapath. Its job is to encapsulate the packet between a Start-Of-Frame (SOF) 16-bit word and a 16-bit Cyclic Redundancy Check (CRC), and transmit the data to the GOL chip, along with the data qualifiers. Figure 3.28 shows the block's schematic.

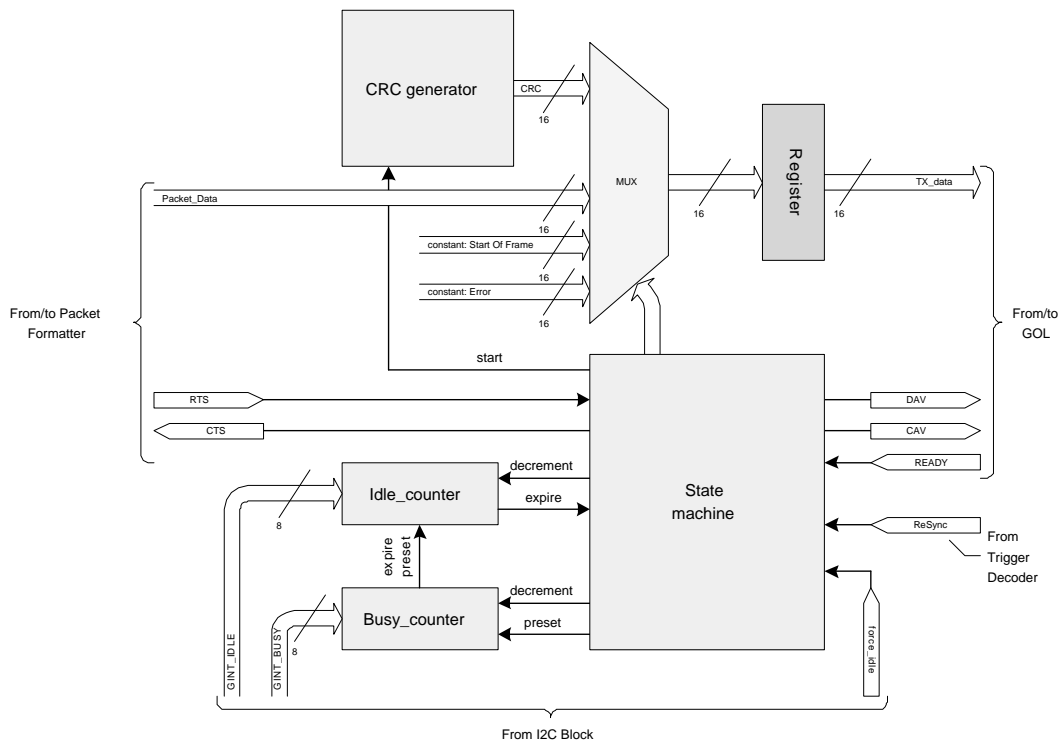


Figure 3.28: GOL Interface schematic.

Since this block deals with the GOL, its two possible protocol mode of *Transmission protocol* operation are here important: the GOL Interface must be capable to handle both of them. The choice has been to drive the GOL with the same identical signals in both the two modes exploiting the characteristics that the two protocol have in common. The GOL Interface is therefore not aware of the chosen GOL mode, since no input signal is giving this information.

Since the Ethernet mode is more limited than the G-Link, this choice constraints the GOL Interface to use only a small set of control commands. These are the Start-Of-Frame and the Error, besides the normal data frames. The two commands are implemented in the G-Link putting a fixed 16-bit value on the data bus, while in the Ethernet mode they are directly mapped to

the available *CarrierExtend* and *ErrorPropagation* command frames. Data words are distinguished from control words using the two data qualifiers DAV and CAV.

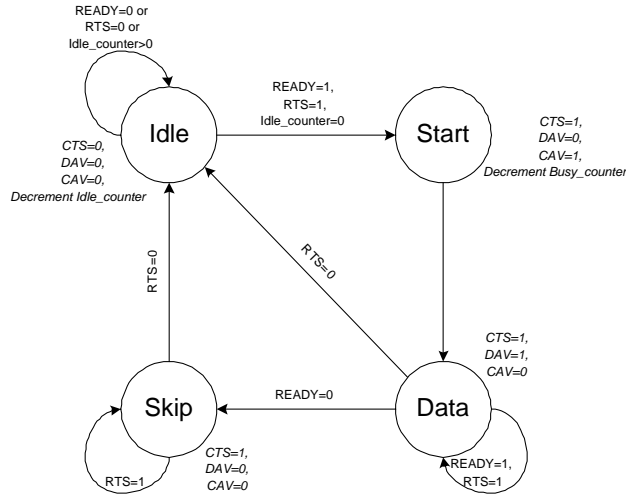


Figure 3.29: GOL Interface state machine diagram.

A simple state machine generates these qualifier signals under the control of the Packet Formatter's RTS and the GOL's READY messages. The state machine's diagram is represented in Figure 3.29. Whenever the RTS is raised, the transmission starts with a SOF command, along with the DAV low and the CAV high. Then the Packet\_Data is sent with DAV high and CAV low. After the last DATA state, the CRC is inserted while going back to the IDLE state. When no data is transmitted, the DAV and CAV remain both low, in order to let the GOL send idle patterns.

The transmission can't be initiated when the READY signal from the GOL is low. If the READY is deasserted during the transmission, the state machine enters in the SKIP state, without notifying to the Packet Formatter the link loss. The current packet is, in this case, lost.

#### Burst transmissions

Burst transmission are possible, and a SOF command is always present between two packets. Figure 3.30 illustrates the timing diagram of a possible transmission.

It is possible to limit the burst length by modifying the two 8-bit I<sup>2</sup>C registers GINT\_BUSY and GINT\_IDLE which are the preset values for two counters embedded in the GOL Interface, respectively the Busy\_counter and the Idle\_counter. In practice, the GOL Interface is forced to insert a number of GINT\_IDLE idle cycles every GINT\_BUSY packets sent. As can be seen from the state diagram, the data is not sent until the Idle\_counter is zero.

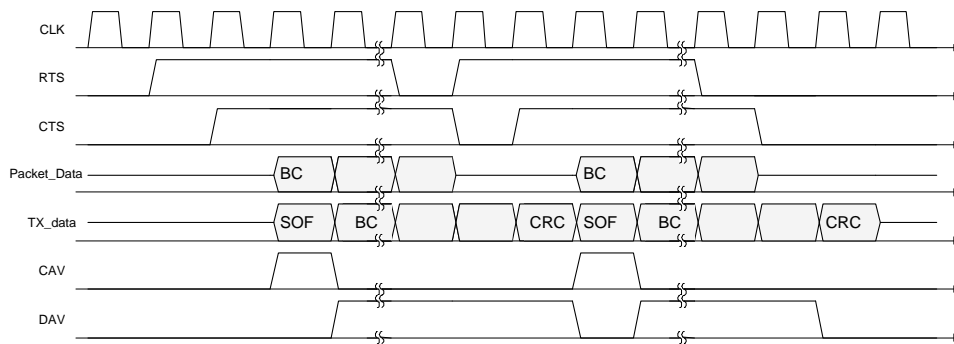


Figure 3.30: GOL Interface timing diagram.

When this happens, the `Busy_counter` is preset, and will be decremented at every packet sent. As soon as the `Busy_counter` expires, a new idle cycle sequence will be inserted by presetting again the `Idle_counter`.

The reason for this complex behavior is that the link might need some idle patterns to be sent in order to maintain the synchronization between the GOL and the receiver in the counting room: although the data frame codings used are themselves redundant since they contain phase synchronization patterns, the idle frames can also recover the link from a complete loss of synchronization. Data frames can't perform this task. Moreover the data link is unidirectional, thus the only communication path with which the counting room can signal link losses is the control token ring, which is not enough fast. The periodical insertion of fill idle frames can prevent link losses.

Another way to insert idle frames is through the `force_idle` input signal, which makes the state machine remain into the `IDLE` state when high. This flag can be set via I<sup>2</sup>C and disables, in practice, the GOL Interface.

The choice to triplicate also the CRC Generator's state machine prevents the use of commercial Intellectual Properties (IPs) for the implementation of this block. A general purpose parameterized CRC generator functional Verilog model has therefore been written, the code can be found in Appendix B. The parameters employed in the Kchip configure the generator to obtain a CCITT standard 16-bit CRC.

The GOL Interface state machine presets the CRC Generator every time it enters in the `START` state.

With the Start-Of-Frame and CRC added the packets sent to the GOL look like Figure 3.31. The normal data packet is 299 words long, and it takes 7.745  $\mu$ s to be transmitted. Therefore the absolute maximum outgoing event rate is 134 kHz.

*CRC Generator*

*Final packet structure*

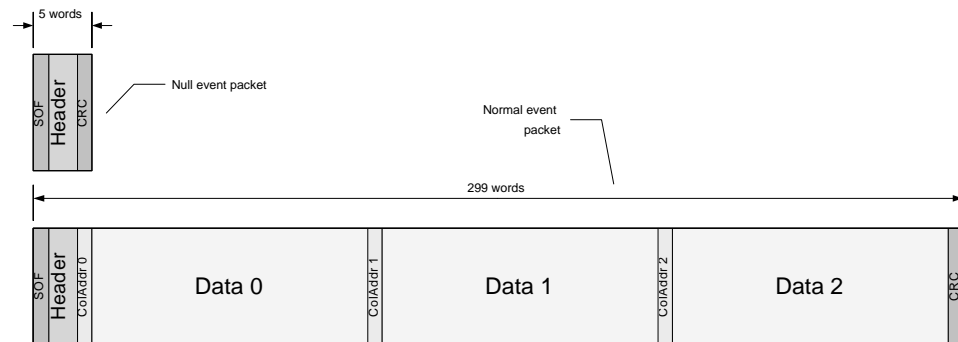


Figure 3.31: Kchip output packets.

### 3.2.16 The CalPulse Builder

The calibration pulses sent to the PACE chips have to respect a few requirements regarding their width and delay in respect to the clock. Figure 3.32 illustrates the block's schematic.

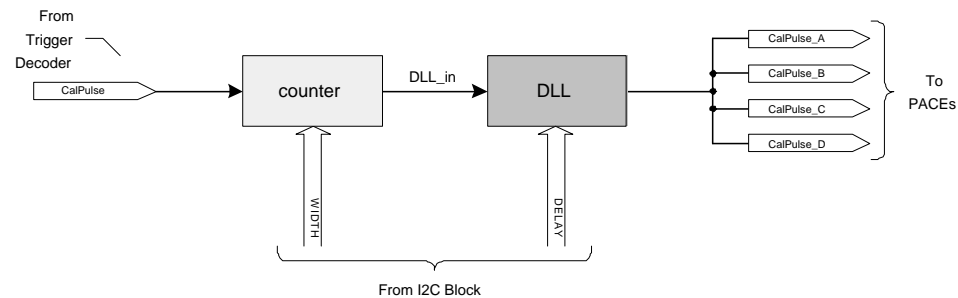


Figure 3.32: CalPulse Builder schematic.

A simple counter, preset to the `WIDTH` value when the `CalPulse` is decoded, takes care of the pulse width: the `DLL_in` signal will be high until the counter is zero. Therefore, the pulse width is programmable with the granularity of one clock cycle.

The Delay Locked Loop<sup>14</sup> (DLL), instead, delays, by a programmable amount of time, its input. In practice the output is an exact copy of the input but delayed. The delay is configurable, with the `DELAY` I<sup>2</sup>C register, in 8 possible steps within a clock cycle.

<sup>14</sup>The basic principles of a Delay Locked Loop can be found in Appendix A. The DLL block is made by Rutherford Appleton Laboratory, in Oxfordshire, UK.

### 3.2.17 The I2C Block

The interface with the I<sup>2</sup>C bus is controlled by the I2C Block which is mainly composed by a synchronizer, a controller and a set of registers with their addressing logic. Figure 3.33 shows the block's schematic.

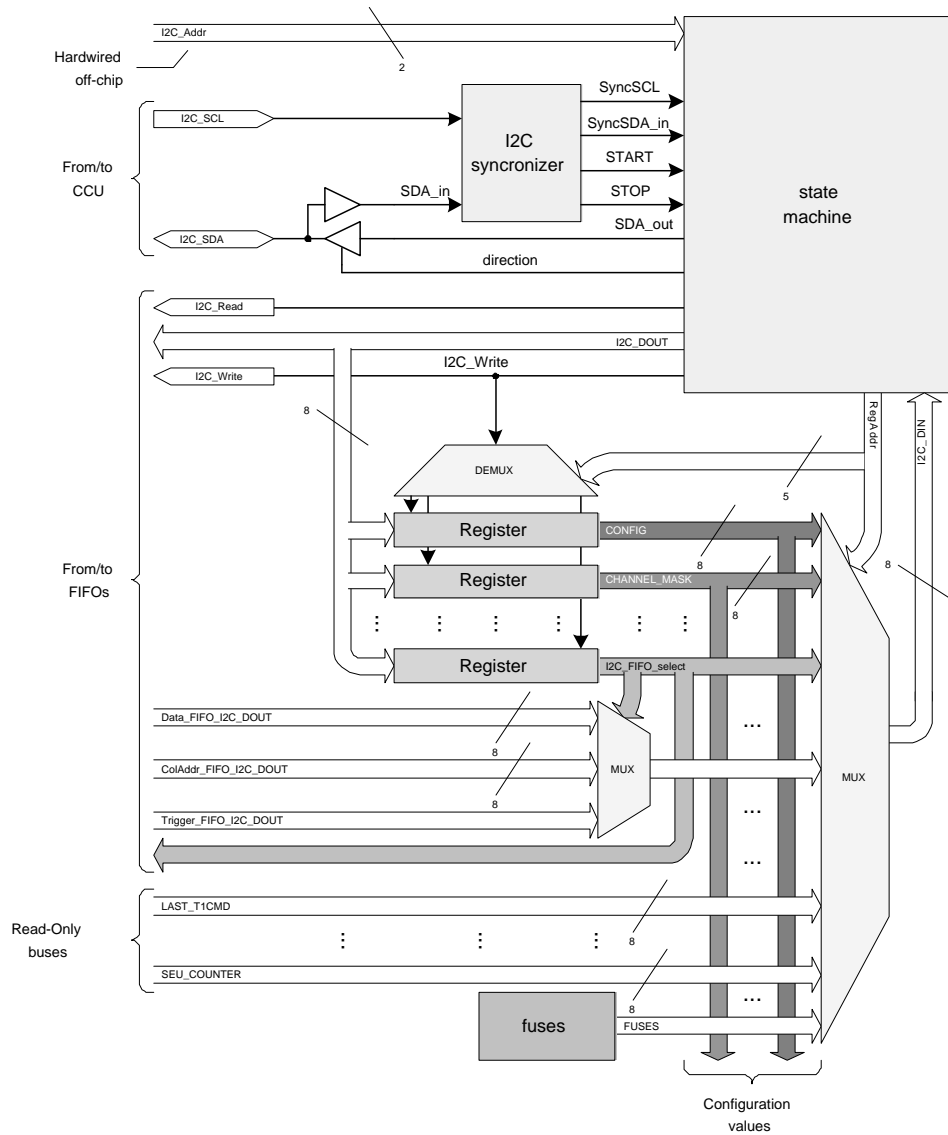


Figure 3.33: I2C Block simplified schematic. Only a few locations in the register addressing space are shown.

The synchronizer is necessary since there is no control on the I2C\_SCL and I2C\_SDA lines delay in respect to the system clock. Metastability prob-

*I2C synchronizer*

lems could arise without a dedicated circuitry. This is resolved by a 2-stages registering of the signals, as shown in Figure 3.34. Possible undefined values sampled by the first flip-flop will end in one of its two allowed stable states within the clock period, and the second flip-flop will get a “clean” signal. The synchronizer also takes care of detecting the beginning and the end of a transaction, decoding the **START** and **STOP** commands.

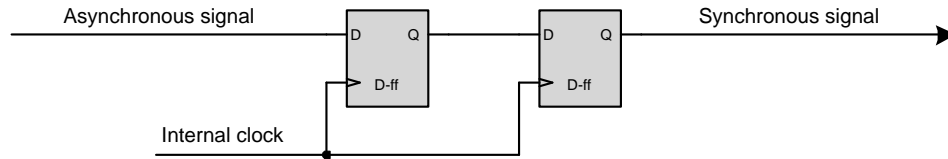


Figure 3.34: Synchronization technique employed in the I2C synchronizer.

### State machine

The state machine (also referred as the I2C Controller) de-serializes the incoming data and performs the opposite operation when transmitting. The de-serialized data is put on the **I2C\_DOUT** bus, while the outgoing comes from the **I2C\_DIN** bus. The Kchip supports only 7-bit-address and 8-bit-data I<sup>2</sup>C transactions (see Section 3.1.7).

The address is split in two parts: a 2-bit device address and a 5-bit register address. Before answering to the master, the 2-bit device address is compared to the actual chip address hardwired on the board, which comes from the **I2C\_Addr** bus. If the device address matches, the communication goes on. The 5-bit register address is instead stored in the **RegAddr** bus: there is space available for 32 registers.

The state machine also controls the direction of the input/output pad, and, on the other side, it creates the **I2C\_Write** and **I2C\_Read** commands which define the operation in progress.

### Register bank

The **RegAddr** is decoded into the necessary select lines, which are combined with the **I2C\_Write** to obtain a set of write enable signals for the registers. At the same time, the **RegAddr** is used to select, through a multiplexer, the right register output bus on the **I2C\_DIN** bus.

The addressing space available is used only partly for read/write configuration registers: some locations are connected to Kchip internal buses for status checking, and are therefore read-only. On top of that, a couple of locations is reserved to access the FIFOs: one is mapped to the MSB, and the other to the LSB of each memory. The memory selection is again possible loading the **I2C\_FIFO\_select** register with the appropriate content.

Table 3.5 describes the mapping of the registers, which have been already been described in the previous chapters.



RegAddr	Register	Access	Bit mapping
0	CONFIG	Read/Write	0-3 CHANNEL_MASK 4 testlink 5 force_idle 6 TriggerInhibitMode 7 KchipMode
1	ECONFIG	Read/Write	0 DLL_off 1 MaskCalLV1 2-7 (unused)
2-3	KID	14 bit R/W, 2 bit R-o	0-1 I2C_Addr 2-15 motherboard ID
4	MASK_T1CMD	Read/Write	0 mask LV1 1 mask ReSync 2 mask CalPulse 3 mask BCO 4-7 (unused)
5	LAST_T1CMD	Read-only	0 LV1 1 ReSync 2 CalPulse 3 BCO 4-7 (unused)
6	LATENCY	Read/Write	
7	EventCounter	Read-only	
8-9	BunchCounter	Read-only	
10	(unused)		
11	GINT_BUSY	Read/Write	
12	GINT_IDLE	Read/Write	
13	I2C_FIFO_select	Read/Write	0-3 Data FIFO 4 Column Addresses FIFO 5 Trigger FIFO 6-7 (unused)
14-15	access to FIFOs	Read/Write	
16-17	STATUS	Read-only	0-5 (unused) 6 READY 7 Disaster 8-11 ErrDataValid 12-15 ErrPACE_FIFO
18	SEU_COUNTER	Read-only	
19	DELAY	Read/Write	
20	WIDTH	Read/Write	
21	ADC_LATENCY	Read/Write	
22	FULL_THRES	Read/Write	
23-29	(unused)		
30-31	FUSES	Read-only	

Table 3.5: I<sup>2</sup>C registers mapping.

*Fuses*

Two special configuration read-only locations are wired to the fuses. This block contains 16 laser-burned fuses, which behave like a small ROM, and contain a unique identifier for the chip among all the manufactured ones. In fact, the fuses will be burnt during the production with an incremental binary configuration.

**3.2.18 Testability**

During the Kchip design, an important issue has been always kept in mind: since the ASIC under development is going to be produced in a number of thousands chips, testing must be easy and relatively fast. This implies several constraints at the HDL description stage like having a smart I2C access to the memories and to other registers, and preparing self-test circuits included in the chip.

Many different kinds of defects may be present at the end of production of an IC: undesired open circuits, shorts, and the most frequent stuck-at-1 and stuck-at-0 [6]. These last two faults identify nets which are charged to a fixed logic value. All these imperfections can lay in every net in the chip, thus finding them is difficult and time consuming, in some cases even impossible, without the aid of a test structure hardwired in the circuit.

What is usually done is to exploit the circuit's registers, which are connected to form a pipeline configuration, loading them with a test pattern and checking their content after 1 clock cycle. Of course it is not possible to wire every register to the outside, in order to load and read it back. A scan chain configuration is therefore used and scan flip-flops are employed instead of normal ones.

Scan flip-flops have a data input line, a scan data input line plus a select input line: in practice, a multiplexer selects between the two data lines and the only the selected one is registered. For each flip-flop, the output is connected to the scan input of another flip-flop, except for the last one, while the select line is shared among all of them. This forms a structure which is shift-register-like in one of the select signal values. The first flip-flop scan input is connected to an input pad, called *test scan input*, and the last flip-flop output goes to an output pad, called *test scan output*. Data can be loaded serially from the outside from the test scan input and read back from the test scan output. A third pad is necessary for the select input line of all the scan flip-flops: this is the *test scan enable* pad. Nevertheless, the test scan input and output pads can be shared with other pads, since their behavior can be determined by the test scan enable value. Figure 3.35 shows an example of scan chain.

In this way a complete test pattern read/write structure is made, adding just one input pad for the test scan enable line<sup>15</sup>. The cost is to use bigger

---

<sup>15</sup>Other techniques employ a second clock tree with clock gating: this approach is more complex and requires more resources.

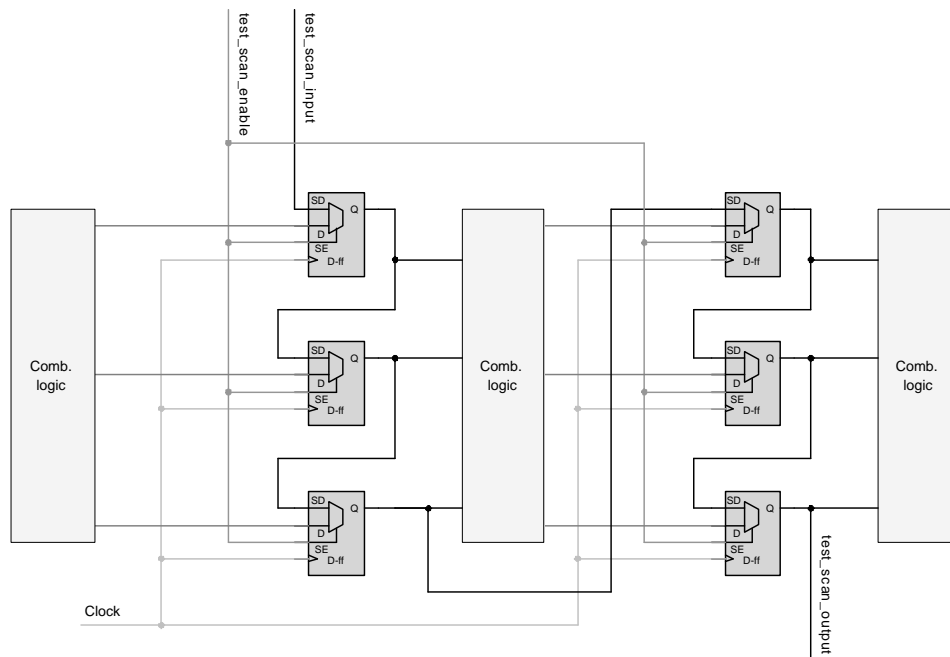


Figure 3.35: Example of pipeline with test scan chain.

flip-flops<sup>16</sup> and route all the connections among them.

Synthesis tools can automatically add a test scan path to an existing design without test structures, and, moreover they can prepare a set of test patterns which cover a large amount of faults for the chip, knowing its netlist. These patterns can then easily be used in production testing.

### 3.2.19 Synthesis

The top module of the hierarchical Verilog description prepared is composed by all the necessary input/output pads and a single instantiation for the core, which is described in another module. Only the core module was synthesized, leaving the pads untouched.

Synthesis is performed leaving only one level of hierarchy to the core and flattening all the logic, in order to leave the maximum freedom to the synthesizer. No clock tree was prepared at this stage, while trees for the **RESETb** and **test\_se** signals were made, since they have a huge load represented by almost all the flip-flops. Table 3.6 resumes the synthesis results.

The tool prepared a test scan-chain excluding the Clock and Control block, the I2C synchronizer and the DeDDR, which contain negative edge flip-flops and would require a dedicated chain. The test scan input is shared

<sup>16</sup>Scan flip-flops use 30% more area than normal ones.

Number of gates	13380	
Estimated standard cell gates area	2.98	mm <sup>2</sup>
Area used for memories	6.22	mm <sup>2</sup>
Area used for other macrocells	0.09	mm <sup>2</sup>
Total area used	9.29	mm <sup>2</sup>
Number of flip-flops	1331	
Number of flip-flops in scan chain	1290	
Number of test patterns created	1777	
Test pattern fault coverage	91	%

Table 3.6: Kchip synthesis results.

with the T1 signal, while the test scan output is on the DAV line. The test scan enable is mapped in the `test_se` signal. A set of test patterns was created.

### 3.2.20 Input/output pads

Most of the signals travelling across the chip boundary are implemented with the Low Voltage Differential Signaling (LVDS) standard. In this way a single-ended internal signal becomes differential outside of the chip (and viceversa), thus a pair of pins is necessary for each LVDS signal. The LVDS technique guarantees low-noise and fast-switching, but the drawback is that the pads require more power, since a current is always circulating through the pins.

The library provides a large set of pads protected against Electrostatic Discharge (ESD), including LDVS input and outputs, single-ended (CMOS) input and outputs, open-collector input/outputs and Schmitt-trigger inputs. Moreover, two different pairs of power pads are available: peripheral power pads and core power pads. This separates the power supply for the core from the supply for the pads themselves, avoiding the switching noise introduced by the pads in the core. Table 3.7 lists all the Kchip pins with the respective pads. The number of power/ground pad pairs has been chosen compatibly with the requirements of the logic and the input/output pads.

The library pads fit into a 125  $\mu\text{m}$  grid: every CMOS pad is 125  $\mu\text{m}$  wide, while every LVDS pad is 250  $\mu\text{m}$  wide. Both the two types of pad occupy 350  $\mu\text{m}$  in the other direction, thus the corner pads are square 350  $\times$  350  $\mu\text{m}^2$ .

When choosing the position of the pads it is important to separate the inputs from the outputs putting a power supply and ground pair in between: this prevents, again, the noise from the output pads, which draw a lot of current, to be introduced in the inputs. The pad position was actually chosen during the floorplanning. The number of single pads on the chip is 151, thus the perimeter of the chip must be at least 21.675 mm.

No.	Pin name	Direction	Type	Internal signal	No.	Pin name	Direction	Type	Internal signal
1	C_ReSync_pos	Output	LVDS+	ReSync_C	76	I2C_addr[1]	Input	CMOS	I2C_addr[1]
2	C_ReSync_neg	Output	LVDS-	ReSync_C	77	I2C_addr[0]	Input	CMOS	I2C_addr[0]
3	D_ReSync_pos	Output	LVDS+	ReSync_D	78	CD_ADC_pos[11]	Input	LVDS+	CD_ADC[11]
4	D_ReSync_neg	Output	LVDS-	ReSync_D	79	CD_ADC_neg[11]	Input	LVDS-	CD_ADC[11]
5	A_CalPulse_pos	Output	LVDS+	CalPulse_A	80	CD_ADC_pos[10]	Input	LVDS+	CD_ADC[10]
6	A_CalPulse_neg	Output	LVDS-	CalPulse_A	81	CD_ADC_neg[10]	Input	LVDS-	CD_ADC[10]
7	B_CalPulse_pos	Output	LVDS+	CalPulse_B	82	CD_ADC_pos[9]	Input	LVDS+	CD_ADC[9]
8	B_CalPulse_neg	Output	LVDS-	CalPulse_B	83	CD_ADC_neg[9]	Input	LVDS-	CD_ADC[9]
9	V <sub>DD</sub>	Power	Core	V <sub>DD</sub>	84	V <sub>DD</sub>	Power	Core	V <sub>DD</sub>
10	GND	Ground	Core	GND	85	GND	Ground	Core	GND
11	C_CalPulse_pos	Output	LVDS+	CalPulse_C	86	CD_ADC_pos[8]	Input	LVDS+	CD_ADC[8]
12	C_CalPulse_neg	Output	LVDS-	CalPulse_C	87	CD_ADC_neg[8]	Input	LVDS-	CD_ADC[8]
13	D_CalPulse_pos	Output	LVDS+	CalPulse_D	88	CD_ADC_pos[7]	Input	LVDS+	CD_ADC[7]
14	D_CalPulse_neg	Output	LVDS-	CalPulse_D	89	CD_ADC_neg[7]	Input	LVDS-	CD_ADC[7]
15	A_LV1_pos	Output	LVDS+	LV1_A	90	CD_ADC_pos[6]	Input	LVDS+	CD_ADC[6]
16	A_LV1_neg	Output	LVDS-	LV1_A	91	CD_ADC_neg[6]	Input	LVDS-	CD_ADC[6]
17	B_LV1_pos	Output	LVDS+	LV1_B	92	CD_ADC_pos[5]	Input	LVDS+	CD_ADC[5]
18	B_LV1_neg	Output	LVDS-	LV1_B	93	CD_ADC_neg[5]	Input	LVDS-	CD_ADC[5]
19	C_LV1_pos	Output	LVDS+	LV1_C	94	CD_ADC_pos[4]	Input	LVDS+	CD_ADC[4]
20	C_LV1_neg	Output	LVDS-	LV1_C	95	CD_ADC_neg[4]	Input	LVDS-	CD_ADC[4]
21	D_LV1_pos	Output	LVDS+	LV1_D	96	GND	Ground	Peripheral	GND
22	D_LV1_neg	Output	LVDS-	LV1_D	97	V <sub>DD</sub>	Power	Peripheral	V <sub>DD</sub>
23	V <sub>DD</sub>	Power	Peripheral	V <sub>DD</sub>	98	CLK_IN_pos	Input	LVDS+	CLK
24	GND	Ground	Peripheral	GND	99	CLK_IN_neg	Input	LVDS-	CLK
25	TX_data[0]	Output	CMOS	TX_data[0]	100	CD_ADC_pos[3]	Input	LVDS+	CD_ADC[3]
26	TX_data[1]	Output	CMOS	TX_data[1]	101	CD_ADC_neg[3]	Input	LVDS-	CD_ADC[3]
27	TX_data[2]	Output	CMOS	TX_data[2]	102	CD_ADC_pos[2]	Input	LVDS+	CD_ADC[2]
28	TX_data[3]	Output	CMOS	TX_data[3]	103	CD_ADC_neg[2]	Input	LVDS-	CD_ADC[2]
29	TX_data[4]	Output	CMOS	TX_data[4]	104	CD_ADC_pos[1]	Input	LVDS+	CD_ADC[1]
30	TX_data[5]	Output	CMOS	TX_data[5]	105	CD_ADC_neg[1]	Input	LVDS-	CD_ADC[1]
31	TX_data[6]	Output	CMOS	TX_data[6]	106	CD_ADC_pos[0]	Input	LVDS+	CD_ADC[0]
32	TX_data[7]	Output	CMOS	TX_data[7]	107	CD_ADC_neg[0]	Input	LVDS-	CD_ADC[0]
33	GND	Ground	Core	GND	108	GND	Ground	Core	GND
34	V <sub>DD</sub>	Power	Core	V <sub>DD</sub>	109	V <sub>DD</sub>	Power	Core	V <sub>DD</sub>
35	TX_data[8]	Output	CMOS	TX_data[8]	110	AB_ADC_pos[11]	Input	LVDS+	AB_ADC[11]
36	TX_data[9]	Output	CMOS	TX_data[9]	111	AB_ADC_neg[11]	Input	LVDS-	AB_ADC[11]
37	TX_data[10]	Output	CMOS	TX_data[10]	112	AB_ADC_pos[10]	Input	LVDS+	AB_ADC[10]
38	TX_data[11]	Output	CMOS	TX_data[11]	113	AB_ADC_neg[10]	Input	LVDS-	AB_ADC[10]
39	TX_data[12]	Output	CMOS	TX_data[12]	114	AB_ADC_pos[9]	Input	LVDS+	AB_ADC[9]
40	TX_data[13]	Output	CMOS	TX_data[13]	115	AB_ADC_neg[9]	Input	LVDS-	AB_ADC[9]
41	TX_data[14]	Output	CMOS	TX_data[14]	116	AB_ADC_pos[8]	Input	LVDS+	AB_ADC[8]
42	TX_data[15]	Output	CMOS	TX_data[15]	117	AB_ADC_neg[8]	Input	LVDS-	AB_ADC[8]
43	DAV	Output	CMOS	DAV	118	AB_ADC_pos[7]	Input	LVDS+	AB_ADC[7]
44	CAV	Output	CMOS	CAV	119	AB_ADC_neg[7]	Input	LVDS-	AB_ADC[7]
45	V <sub>DD</sub>	Power	Peripheral	V <sub>DD</sub>	120	AB_ADC_pos[6]	Input	LVDS+	AB_ADC[6]
46	GND	Ground	Peripheral	GND	121	AB_ADC_neg[6]	Input	LVDS-	AB_ADC[6]
47	READY	Input	CMOS	READY	122	AB_ADC_pos[5]	Input	LVDS+	AB_ADC[5]
48	A_DataValid_neg	Input	LVDS-	DataValid_A	123	AB_ADC_neg[5]	Input	LVDS-	AB_ADC[5]
49	A_DataValid_pos	Input	LVDS+	DataValid_A	124	AB_ADC_pos[4]	Input	LVDS+	AB_ADC[4]
50	A_ColAddr_neg	Input	LVDS-	ColAddr_A	125	AB_ADC_neg[4]	Input	LVDS-	AB_ADC[4]
51	A_ColAddr_pos	Input	LVDS+	ColAddr_A	126	AB_ADC_pos[3]	Input	LVDS+	AB_ADC[3]
52	A_FIFO_Full	Input	CMOS	FIFO_Full_A	127	AB_ADC_neg[3]	Input	LVDS-	AB_ADC[3]
53	B_DataValid_neg	Input	LVDS-	DataValid_B	128	AB_ADC_pos[2]	Input	LVDS+	AB_ADC[2]
54	B_DataValid_pos	Input	LVDS+	DataValid_B	129	AB_ADC_neg[2]	Input	LVDS-	AB_ADC[2]
55	B_ColAddr_neg	Input	LVDS-	ColAddr_B	130	AB_ADC_pos[1]	Input	LVDS+	AB_ADC[1]
56	B_ColAddr_pos	Input	LVDS+	ColAddr_B	131	AB_ADC_neg[1]	Input	LVDS-	AB_ADC[1]
57	B_FIFO_Full	Input	CMOS	FIFO_Full_B	132	AB_ADC_pos[0]	Input	LVDS+	AB_ADC[0]
58	C_DataValid_neg	Input	LVDS-	DataValid_C	133	AB_ADC_neg[0]	Input	LVDS-	AB_ADC[0]
59	C_DataValid_pos	Input	LVDS+	DataValid_C	134	GND	Ground	Peripheral	GND
60	C_ColAddr_neg	Input	LVDS-	ColAddr_C	135	V <sub>DD</sub>	Power	Peripheral	V <sub>DD</sub>
61	C_ColAddr_pos	Input	LVDS+	ColAddr_C	136	ADC_CLK_neg	Output	LVDS-	ADC_CLK
62	C_FIFO_Full	Input	CMOS	FIFO_Full_C	137	ADC_CLK_pos	Output	LVDS+	ADC_CLK
63	V <sub>DD</sub>	Power	Peripheral	V <sub>DD</sub>	138	A_PACE_CLK_neg	Output	LVDS-	PACE_CLK_A
64	GND	Ground	Peripheral	GND	139	A_PACE_CLK_pos	Output	LVDS+	PACE_CLK_A
65	D_DataValid_neg	Input	LVDS-	DataValid_D	140	B_PACE_CLK_neg	Output	LVDS-	PACE_CLK_B
66	D_DataValid_pos	Input	LVDS+	DataValid_D	141	B_PACE_CLK_pos	Output	LVDS+	PACE_CLK_B
67	D_ColAddr_neg	Input	LVDS-	ColAddr_D	142	C_PACE_CLK_neg	Output	LVDS-	PACE_CLK_C
68	D_ColAddr_pos	Input	LVDS+	ColAddr_D	143	C_PACE_CLK_pos	Output	LVDS+	PACE_CLK_C
69	D_FIFO_Full	Input	CMOS	FIFO_Full_D	144	D_PACE_CLK_neg	Output	LVDS-	PACE_CLK_D
70	RESEtb	Input	CMOS S-T	RESEtb	145	D_PACE_CLK_pos	Output	LVDS+	PACE_CLK_D
71	T1_neg	Input	LVDS-	T1	146	GND	Ground	Peripheral	GND
72	T1_pos	Input	LVDS+	T1	147	V <sub>DD</sub>	Power	Peripheral	V <sub>DD</sub>
73	test_se	Input	CMOS	test_se	148	A_ReSync_neg	Output	LVDS-	ReSync_A
74	I2C_SCL	Input	CMOS	I2C_SCL	149	A_ReSync_pos	Output	LVDS+	ReSync_A
75	I2C_SDA	Bidir.	CMOS O-C	SDA_in, SDA_out	150	B_ReSync_neg	Output	LVDS-	ReSync_B
					151	B_ReSync_pos	Output	LVDS+	ReSync_B

Table 3.7: Kchip pin-out (note: “S-T” stays for Schmitt-Trigger, and “O-C” for Open-Collector). Pins are numbered counterclockwise in the layout, starting from the corner logo. The horizontal double lines on the table divide the four sides of the chip.

### 3.2.21 Floorplanning

Before starting the Place & Route tool it is necessary to plan the physical structure of the chip. This stage is referred as *floorplanning*.

As results from the synthesis report, the area necessary to accommodate all the cells is about  $10 \text{ mm}^2$ . However the design is pad-limited: in other words the number of pads requires the perimeter of the chip to be larger than what is needed by the core. Another important consideration is that the Kchip will be manufactured together with the PACE chips and the ADCs, meaning in the same wafers: it is therefore necessary that the chips fit into a reticle in order to let the dicing to be possible. The best solution found is for a  $6 \times 5 \text{ mm}^2$  chip, which has a perimeter of 22 mm, enough to fit all the pads.

#### *Core*

The standard cell rows are placed in the center of the chip forming approximately a  $2 \times 2 \text{ mm}^2$  square, which is more than enough to accommodate all the logic plus the clock tree: in fact the clock tree will be created after the placement of the standard cells, therefore it is necessary to reserve space for its buffers at this stage. When working with non-radiation-hard standard cell libraries it is usually not possible to abut the standard cell rows together to compact the area used by the core, since some routing space has to be left among the gates. Even though, the radiation tolerant library's cells are bigger than the normal ones, thus this additive routing space is not mandatory: the rows are therefore placed one next to the other with no space in between. Every row is  $16 \mu\text{m}$  thick.

#### *Pads*

Pads are placed respecting the considerations described in the previous section, and also trying to keep together pins related to the same circuitry: this will help during the placement of the cells.

#### *Macrocells*

The macrocells are placed manually around the standard cells, trying to keep a small distance from them in order to decrease the interconnections' length. The space for power and signal routing is anyway left around the core. Macrocells related to specific inputs or outputs are placed closely to them as much as possible.

#### *Power routing*

Power routing is done with  $100 \mu\text{m}$  thick lines going around the core and around each macrocell. The core power ring is then connected to the 4 power/ground pad pairs. The metal layers used in the ring are level-2 and level-3, thus less resistive (per area unit).

A set of 5 power/ground pairs of vertical stripes crosses through all the standard cells, equally spaced. Every row has its own power lines going horizontally through the core, and these are connected to the vertical stripes at cross-overs.

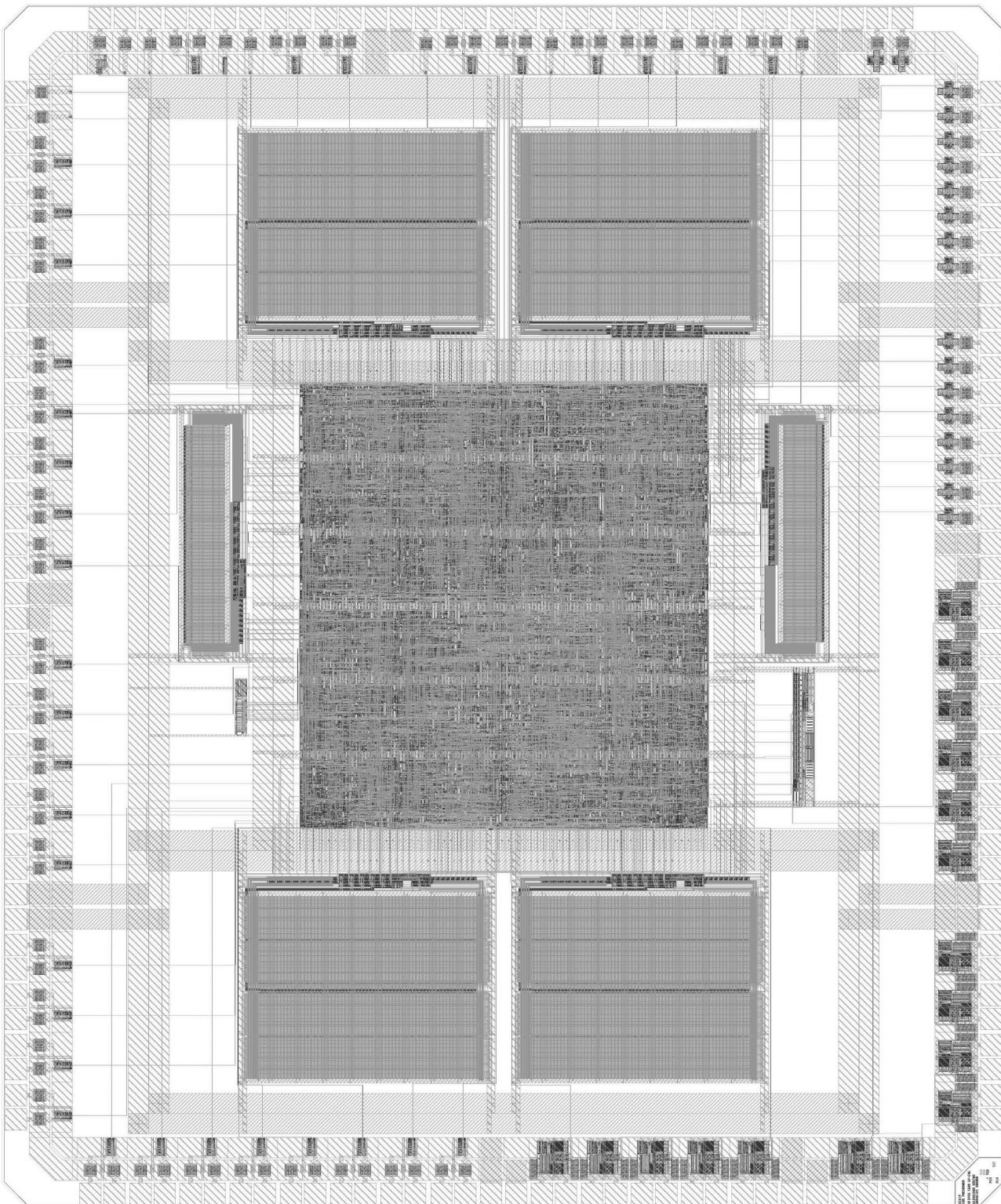


Figure 3.36: Kchip final layout. Figure is rotated 90 degrees counterclockwise.

### 3.2.22 Place & Route

After the floorplanning the automatic placement of the cells is performed, trying to minimize the distance between connected gates.

The clock tree generation is done optimizing the clock skew: the result is a maximum skew of less than 50 ps and a maximum delay, from the root, of  $\approx 700$  ps in typical conditions<sup>17</sup>. To reach this result the clock routing is also balanced.

Signal routing is again done automatically minimizing the interconnection lengths in order to decrease their capacitance. The router was constrained to use only 3 metal layers: this lowers the costs for the production, since less masks are needed.

Figure 3.36 illustrates the final layout (The Figure is rotated 90 degrees counterclockwise, the bottom of the layout is at the right of the page). The four 1 Kword memories are the big blocks at the left and right sides of the layout, while the other smaller memories are placed on the other two sides. The Trigger FIFO SRAM is at the top, close to the fuses, while the Column Addresses FIFO SRAM is at the bottom near the DLL.

The corner logo with the names of the authors is at the bottom left.

## 3.3 The Kchip prototype

A prototype has been designed for the Kchip: this is the KchipB, a smaller version of the circuit which keeps only part of the functionalities. The chip was constrained to fit in a  $3.15 \times 2.00$  mm<sup>2</sup> area. The substantial differences between the Kchip and the KchipB are:

- Only one of the four input channels is kept: this reduces the number of necessary pads. The three unused channels are hardwired to zero;
- The Column Addresses and Data FIFOs are sized to contain just 1 event, which can be possibly loaded through the I<sup>2</sup>C interface. The Data FIFO is implemented with a  $128 \times 27$  bit memory macrocell. The Column Address FIFO, instead, is made out of D flip-flops forming a  $16 \times 8$  bit memory;
- The Trigger FIFO is as well smaller and made out of D flip-flops: it's a  $18 \times 8$  bit memory;
- Since only 1 PACE can be connected to the chip, the PACE Controller doesn't need to supervise the synchronization among the PACEs, and no DataValid and PACE\_full signals are produced internally: the single signals coming from outside are used. For the same reasons the Error Logger is simpler and is limited to its SEU counting function;

---

<sup>17</sup>Typical conditions are temperature  $T = 25$  °C, power supply  $V_{DD} = 2.5$  V, nominal process.



- The use of LVDS pads is reduced to just the clock inputs and outputs;
- No test scan chain is present.

The chip has been synthesized, placed and routed automatically. The final layout contains 58 pads, thus after production it has been bonded in a PGA-100 package. The layout with the indicated pin out can be seen in Figure 3.37, together with a microscope photograph of the chip.

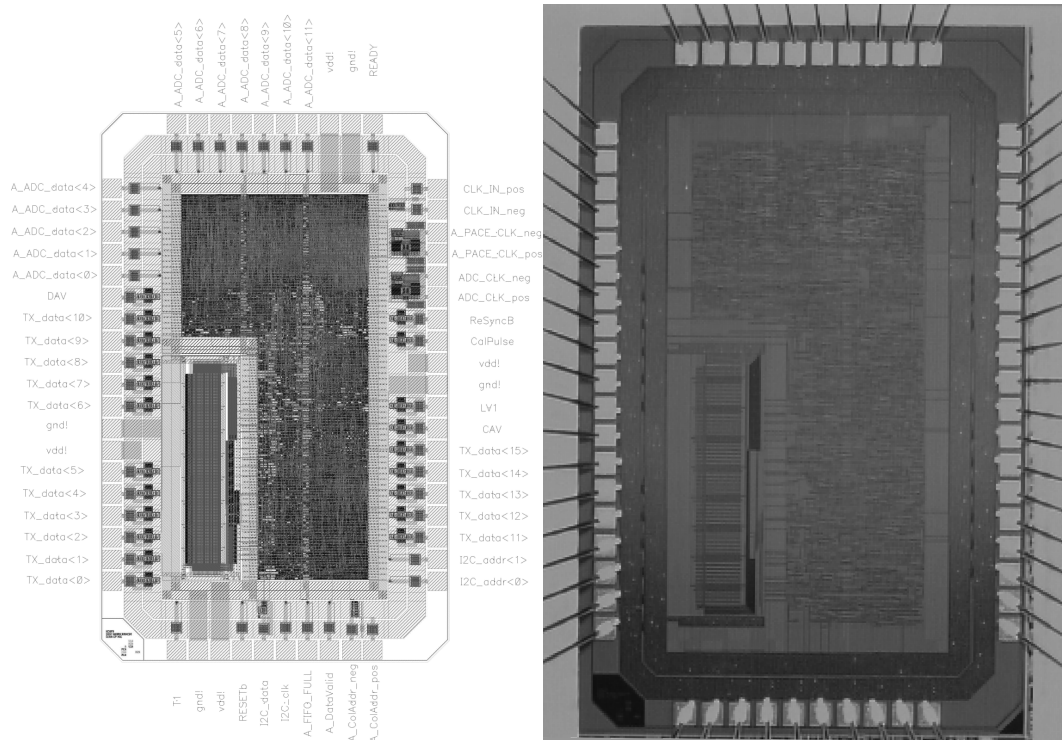


Figure 3.37: KchipB final layout (left) and microscope photograph of the packaged chip (right).

Two sets of test pattern has been prepared converting the data from a few Verilog simulations: these check the capability of writing and reading through I<sup>2</sup>C and of reading out a few events, using random input data. Testing has been done using a digital tester which can apply the input sequence to the chip and cross check the outputs versus their expected value.

Up to now, only 2 chips were tested, obtaining the Shmoo plot shown in Figure 3.38, that should be considered anyway as preliminary. The chips are able to work in nominal conditions as well as down to 2.3 V at nominal frequency, and up to 57 MHz at nominal power supply voltage.

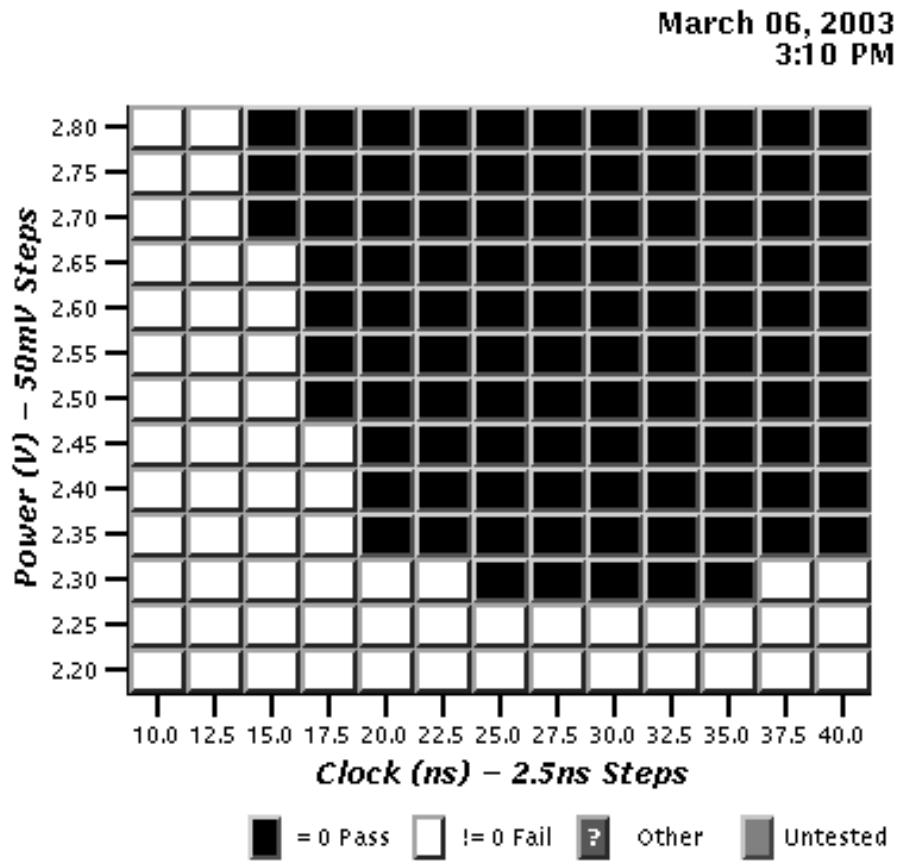


Figure 3.38: PRELIMINARY — KchipB Shmoo plot showing the testing results. Test is performed for each one of the tiles in the plot, obtaining a failure (white) or a success (black), and evaluating different conditions of power supply voltage  $V_{DD}$  and clock frequency.

## Chapter 4

# A radiation tolerant CMOS 0.13 micron Static RAM

The possibility of implementing the Kchip in a 0.13  $\mu\text{m}$  technology will be soon available, therefore a second job I had assigned regarded the full-custom design of a Static Random Access Memory (SRAM) in the relatively new 0.13  $\mu\text{m}$  CMOS technology. Radiation tolerance studies on this technology are currently being performed and the SRAM itself serves also for SEU sensibility testing. On top of that, a recent small standard cells radiation tolerant library has been prepared<sup>1</sup>, but it is not characterized yet.

The 0.13  $\mu\text{m}$  SRAM is based on a previous design in the 0.25  $\mu\text{m}$  technology [20], retaining the same architecture and goals.

The first design done, which is presented here, is a small  $256 \times 9$  bit memory, but, since the layout has a scalable architecture, bigger sizes can be easily obtained: the dimension of the present design was actually fixed by area constraints.

### 4.1 Architecture

The purpose of this design is to obtain a dual-ported size-configurable SRAM suitable for being embedded in radiation tolerant ASICs. To minimize the macrocell area a single-port memory cell is employed and dual-port operation is reached using the first half of the clock cycle for reading, while the second half for writing. Of course the speed of the memory will be limited by this choice, but at the same time the area gain will be high, since a single-port memory cell is much smaller than a dual-port one. On top of that, the speed requirements are not extreme, since many applications run on a slow clock, and, on the other hand, area occupancy affects directly the cost for a chip, thus in many cases is worth to gain area by losing speed.

---

<sup>1</sup>The CMOS 0.13  $\mu\text{m}$  radiation tolerant library has been designed by Kurt Hansler and Robert Szczygiel, CERN EP/MIC.

The scalability of the SRAM is accomplished by its modularity and by the use of self-timed control circuits. The SRAM is in fact subdivided in blocks that can be replicated and abutted to form the desired memory size. The timing circuitry is designed to adapt itself to the size of the memory: the delays are not fixed but vary intrinsically with the scaling of the SRAM.

## 4.2 The memory cell

The SRAM is made out of classic 6-transistor cells, with 2 cross-coupled inverters and 2 access pass-transistor. Figure 4.1 illustrates the basic SRAM cell.

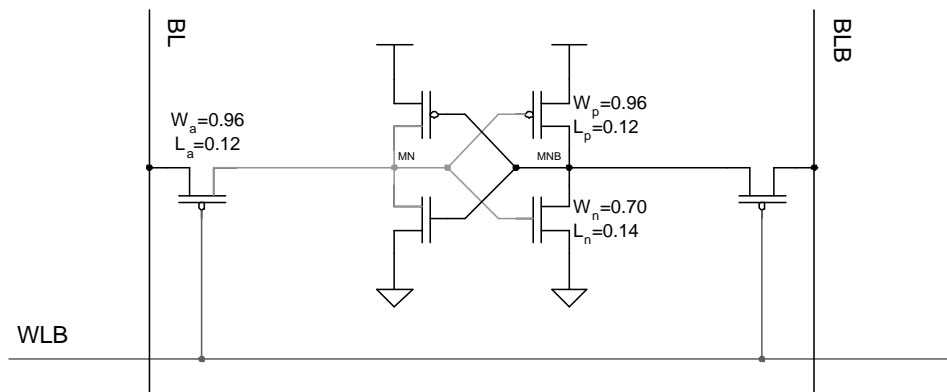


Figure 4.1: SRAM cell schematic. Measures are expressed in microns.

Every cell has a word-line (WLB) and two logically opposite bit-line connections (BL and BLB). The word-lines are connected to the pass-transistors' gates, controlling in this way the access of the bit-lines to the memory nodes (MN and MNB). The two identical inverters constitute in fact the memory unit, since they are wired in a bistable configuration.

It is possible to notice that the pass-transistors are p-channel, in contrast with the usual n-channel used in almost all of the commercial memories. This is because an n-channel transistor has to be Enclosed Layout when following a radiation tolerant approach, while a p-channel doesn't, requiring therefore less area although it needs to have a bigger W/L ratio.

Using p-channel pass-transistor implies the word-line to be low-active: the electrical connection between the bit-lines and the memory nodes is established when WLB is zero. Therefore the word-line will be low only when reading or writing to the cell. For the same reason, before each write operation the bit-lines will be pre-discharged to ground rather than pre-charging them to  $V_{DD}$ : the p-channel transistor can pull-up very well but give worse

performance in pulling down<sup>2</sup>.

#### 4.2.1 Sizing the cell

For area occupancy reasons, the sizes of the transistors is kept as small as possible. Some criteria have anyway to be respected: for example, the two inverters need to have a threshold  $V_{th}$  very close to half of the supply voltage  $V_{DD}$  and a high gain when biased at the threshold. This is compulsory in order to have a good noise immunity during storage: noise coupled to the memory nodes MN and MNB could possibly corrupt the cell content. Charged particles crossing through the silicon can be thought as noise that can produce SEUs.

The threshold can be shifted up or down adjusting the  $r = \frac{W_p/L_p}{W_n/L_n}$  value which expresses the shape factor ratio of the two transistor of each inverter. If the p-channel transistor is stronger than the n-channel, the threshold is lower than  $V_{DD}/2$ , and viceversa. The best solution, taking into account the technology characteristics, is  $r \approx 3.4$ , but this requires a small n-channel transistor, and a big p-channel transistor.

*Inverter sizing*

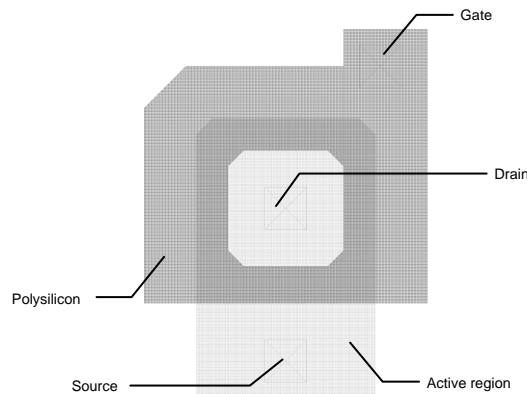


Figure 4.2: Enclosed Layout Transistor using a special technique to obtain narrow channel. The represented n-channel transistor is used in the SRAM cell. The drain contact is at the center of the figure, while the source is at the bottom and the gate contact is at the top.

Using the radiation-tolerant approach doesn't allow to have an arbitrary small n-transistor: a special layout technique is used in the SRAM cell. Figure 4.2 illustrates a detail of the cell layout, where a narrow ELT is drawn. As can be seen, in respect to a traditional ELT, the gate is exceeding

<sup>2</sup>The best read performance could be obtained with pre-charging the bit-lines at  $V_{DD}/2$ , but this requires a voltage generation circuit which draws power and area.

the active region in three directions, leaving only one side for the source. This layout style was first introduced in [11].

The leakage path, dangerous in radiation environment, which goes from the drain to the source under a thick oxide, is again eliminated, since the drain is surrounded by the gate overlapping the active region. The minimum dimensions of the object are fixed by the technology design rules, and are  $L_{min} = 0.12 \mu\text{m}$  and  $W_{min} = 0.68 \mu\text{m}$ . However a smaller  $W_n/L_n = 5$  is obtained with the parameters indicated in the schematic: this allows to make also the p-channel transistor smaller. The chosen shape factor ratio is therefore  $r = 1.6$  that guarantees a threshold voltage  $V_{th} = 0.44 V_{DD}$ .

A qualitative analysis of the memory cell can be done plotting the DC sweep input/output characteristic of the inverter used in the cell, together with its 45-degrees mirror curve: this is the so called *butterfly diagram*, shown in Figure 4.3.

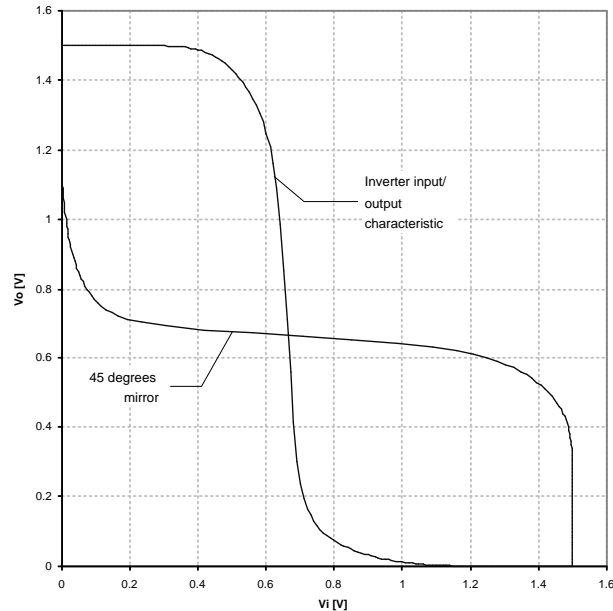


Figure 4.3: Butterfly diagram. DC analysis is plotted with  $V_{DD} = 1.5 \text{ V}$ .

The more the two curves are far away one to the other, the more the cell will be immune to noise [21].

#### *Access transistor size*

The pass-transistors must be enough strong to let a read/write operation to complete successfully: a wider transistors have a smaller resistivity, thus they guarantee a better access to the memory. On the other hand the transistor area should be small to keep the SRAM cell density high.

The best solution is to make the p-channel pass-transistors as wide as the inverters' p-channel MOS: this gives an excellent layout together with a good access capability.

#### 4.2.2 SPICE simulations

SPICE simulations were run to check the cell behavior and its read/write performance. Here two of them will be highlighted: a read test and a write test.

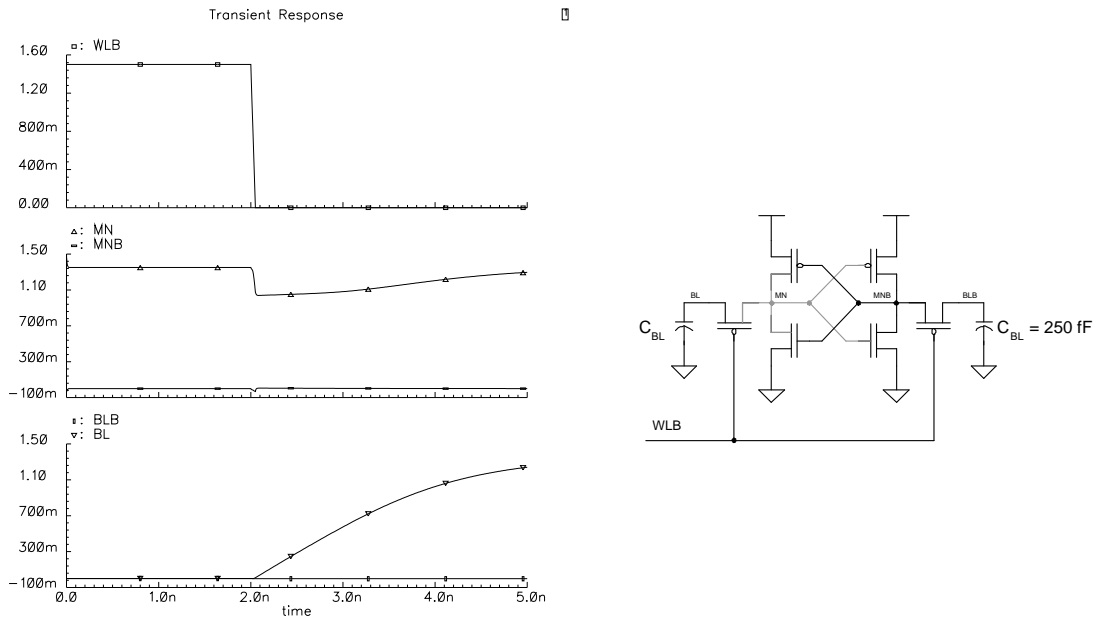


Figure 4.4: SPICE read simulation. On the left, the result is shown, while on the right the circuit used is illustrated.

For the read test, the bit-line capacitance was estimated<sup>3</sup> and found to be  $C_{BL} \simeq 227$  fF. The circuit used for simulation contains therefore two capacitances in place of the bit-lines connected to the SRAM cell: an initial condition of zero voltage on both the bit-lines was applied, since the bit-lines are pre-discharged before reading, thus  $V_{BL}(0) = V_{BLB}(0) = 0$ . Moreover, an initial '1' was stored in the cell:  $V_{MN}(0) = V_{DD}$ ,  $V_{MNB}(0) = 0$ . The word-line input, high at the beginning, is de-asserted after a few nanoseconds. Figure 4.4 shows the simulation results.

As can be seen, there is a slight variation, of about 31 mV, on the memory node MN voltage when the read operation begins, but this does not harm the

<sup>3</sup>Estimation was done counting the metal interconnection capacitance, that is 117 fF for a  $\approx 500$   $\mu\text{m}$  long metal-2 bit-line, as well as the drain diffusion capacitance of the pass-transistors connected, which is 0.857 fF for each transistor.

memory content. The bit-line BL is carried upwards to the logic threshold within 1.3 ns after the word-line is de-asserted.

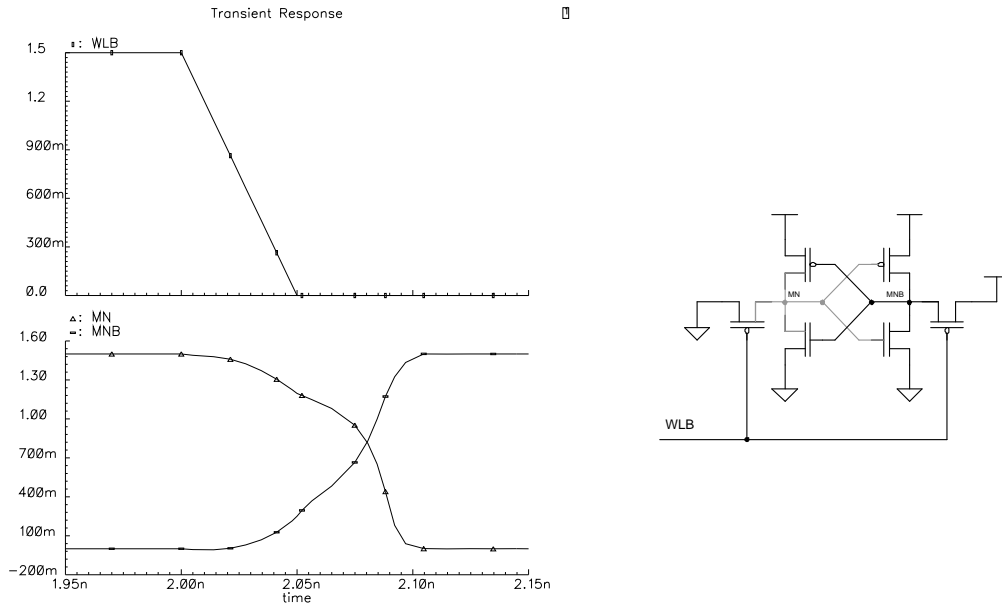


Figure 4.5: SPICE write simulation. On the left, the result is shown, while on the right the circuit used is illustrated.

In the write simulation, instead, one of the bit-line was forced to ground, while the other to  $V_{DD}$ , trying to store a ‘0’ in the cell. An opposite value was previously memorized. Figure 4.5 shows the simulation results: the time taken to write the memory is very short, about 70 ps.

### 4.2.3 Cell layout

The cell’s final layout is represented in Figure 4.6. The two vertical bit-lines are routed in the second metal layer (metal-2) while the horizontal word-line is made of the first metal layer (metal-1) which is more resistive. However, as will be seen later, the word-line travels a short distance compared to the bit-line.

In the middle of the cell lays the vertical power rail in metal-2, which is connected to the horizontal metal-1 line at the top wiring the source of the p-channel inverter transistors. The ground rails are instead on the left and right sides and at the bottom, wiring the source of the n-channel transistors. Not shown in the Figure, two additive power/ground horizontal rails are present, routed in metal-3.

The cell can be replicated as many times as the number of bits desired. Adjacent cells will share the power rails and the guard-rings, maximizing



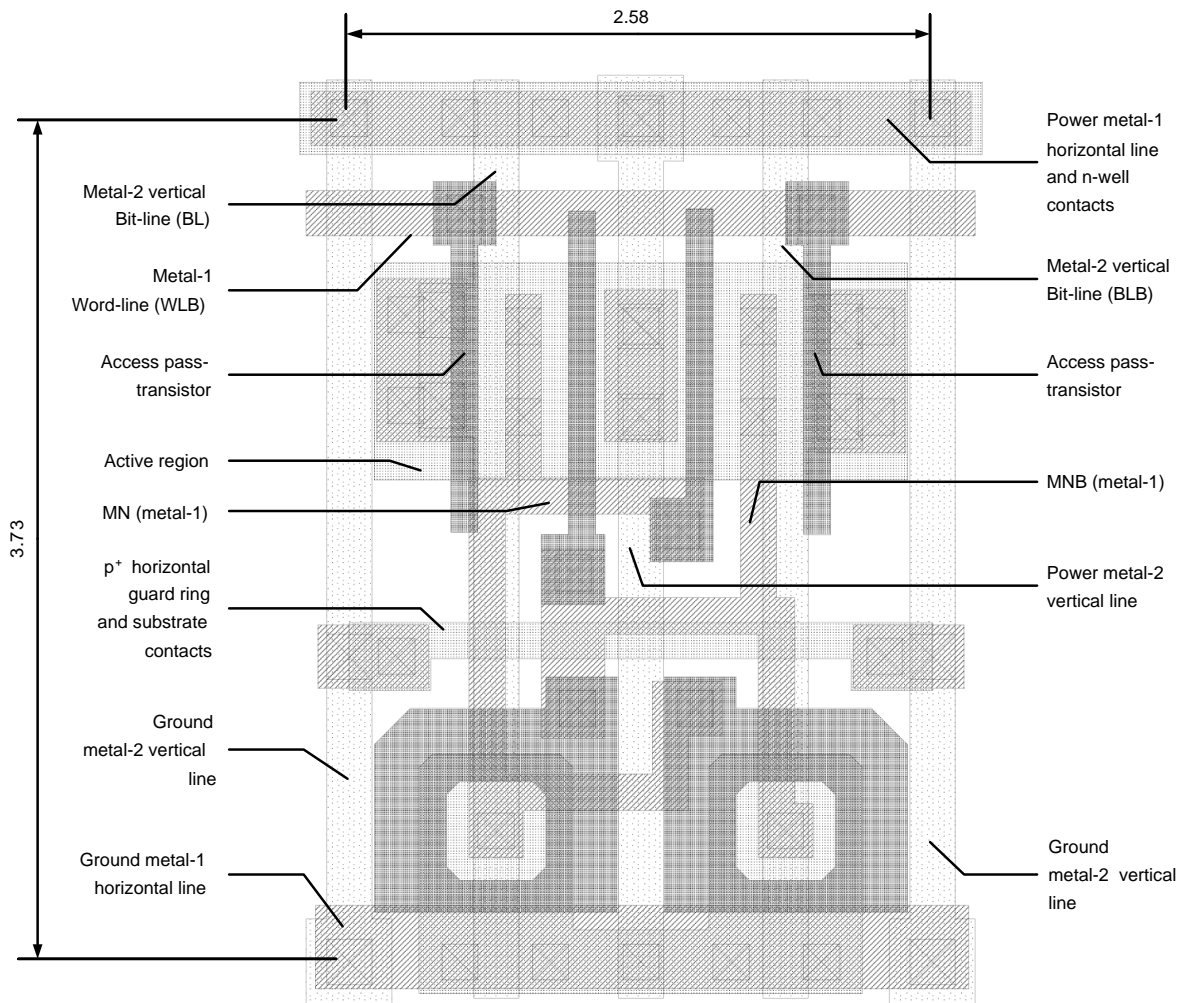


Figure 4.6: SRAM cell layout. Measures are expressed in microns.

the density. The effective sizes of the cell are therefore  $2.58 \times 3.73 \mu\text{m}^2$  obtaining a density of  $104 \text{ kbit}/\text{mm}^2$ , an improvement of 5 times in respect to the previously existing quarter-micron technology SRAM.

### 4.3 The SRAM core

A  $129 \times 19$  array of cells constitutes the memory block. The last row at the bottom and the last column of cells on the right of the memory are not used for data storage but for timing purposes: they are in fact crossed respectively by the dummy word-line (WLBdummy) and the dummy bit-lines

(BLdummy and BLBdummy). Their behavior will be explained later.

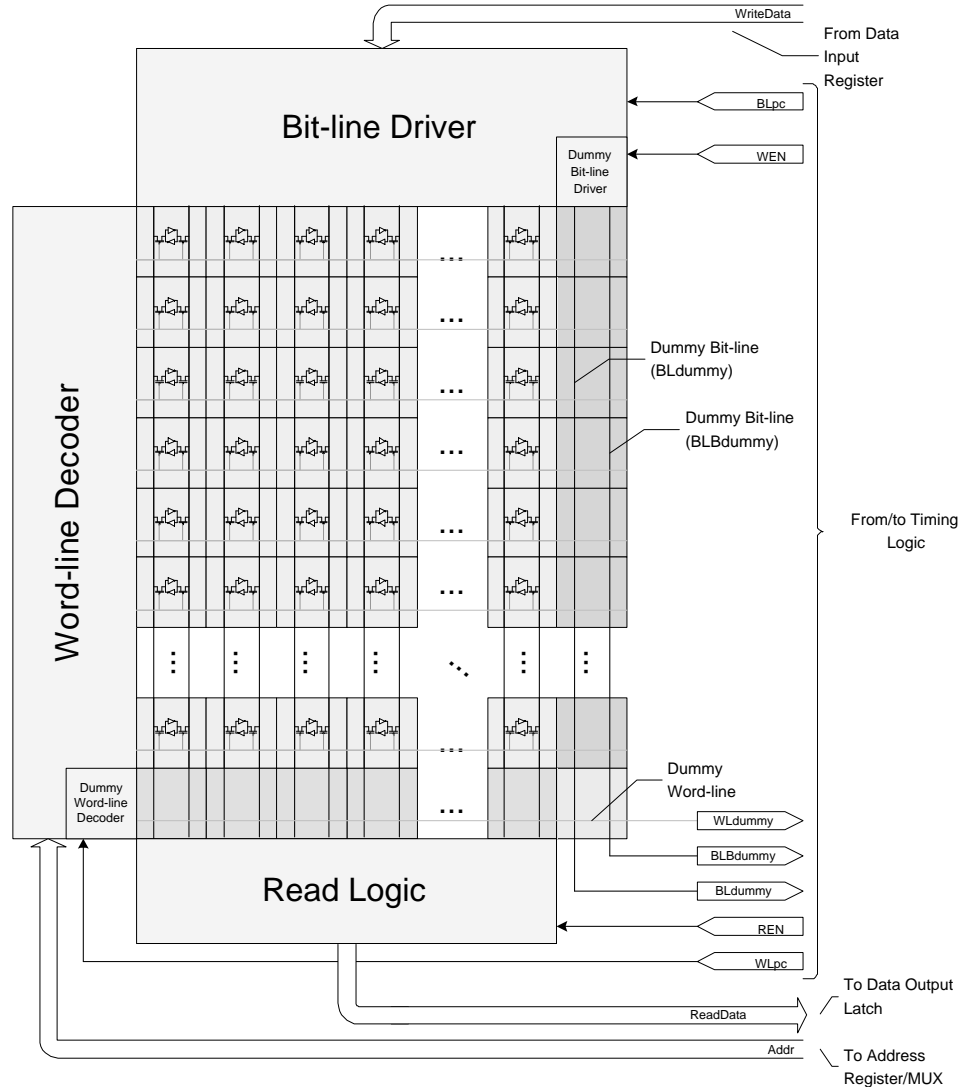


Figure 4.7: SRAM core schematic.

When the memory is idle, the word-lines are all pre-charged to  $V_{DD}$  while the bit-lines are all pre-discharged to ground. As soon as a read or write operation on a cell has to take place, the respective word-line is de-asserted, and all the access transistor on that row will be conductive. A **Word-line Decoder**, laying on the left side of the array, takes care of driving the right word-line low.

Then, in case of a read operation, the bit-lines will be driven by the

cells in the selected row to the stored values. A simple **Read Logic** circuitry, placed on the bottom of the array, will sense the bit-lines and output the data. In case of a write operation, instead, a **Bit-line Driver**, laying on the top of the array, will strongly keep the bit-lines at the logic levels desired, storing the data into the selected cells.

The memory array, together with the **Word-line Decoder**, the **Bit-line Driver** and the **Read Logic**, form the **SRAM core**, which is represented in Figure 4.7. All the blocks in the SRAM core were designed full-custom, since their size depends on the memory cell size. The **memory array** is logically divided into two 9-bit wide blocks, obtaining the mentioned  $256 \times 9$  capacity.

### 4.3.1 The Word-line Decoder

The **Word-line Decoder** is made out replicating 129 times the same basic cell, which is illustrated in Figure 4.8. The basic cell is in practice a pseudo-dynamic 7-input OR: the first stage is what looks like a dynamic NOR gate, while the second stage inverts the output and, at the same time, through a second weak inverter, prevents the dynamic node to be in a high impedance condition. Each basic cell receives a 7-bit binary combination of the address and its negated value, forming a progressive numbering of the lines. The eight address bit is not used by the decoder but outside of the core, to select between the two memory blocks.

The address bus, together with its negation, travels vertically along the decoder on metal-1 lines. The same applies to the pre-charge signal **WLpc**. The driver is asymmetric: pre-charging is faster than de-asserting the word-line. This last task takes about  $t_{WL} \simeq 400$  ps, depending on the operating conditions.

The dummy word-line has an identical decoder connected, but its OR inputs are all hardwired to ground: it will be de-asserted as soon as the phase **WLpc** goes to zero. Last, but not least, the physical vertical size of the cell is the same as the one of the memory cell: this constrains a lot the layout, and the resulting cell is very wide (about  $15 \mu\text{m}$ ).

### 4.3.2 The Bit-line driver

The **Bit-line driver** is as well a replica of the a basic cell, represented in Figure 4.9, per each bit-line. The block receives the data to be written both in its positive and negated form and each basic cell is connected to the proper signal in the bus. The basic cell is just a very strong C<sup>2</sup>MOS inverter plus a pre-discharge transistor. The inverter is enabled by the **WEN** signal: since there are two 9-bit blocks in the memory but only one has to be written per time, the two blocks' drivers receive a different write enable signal.

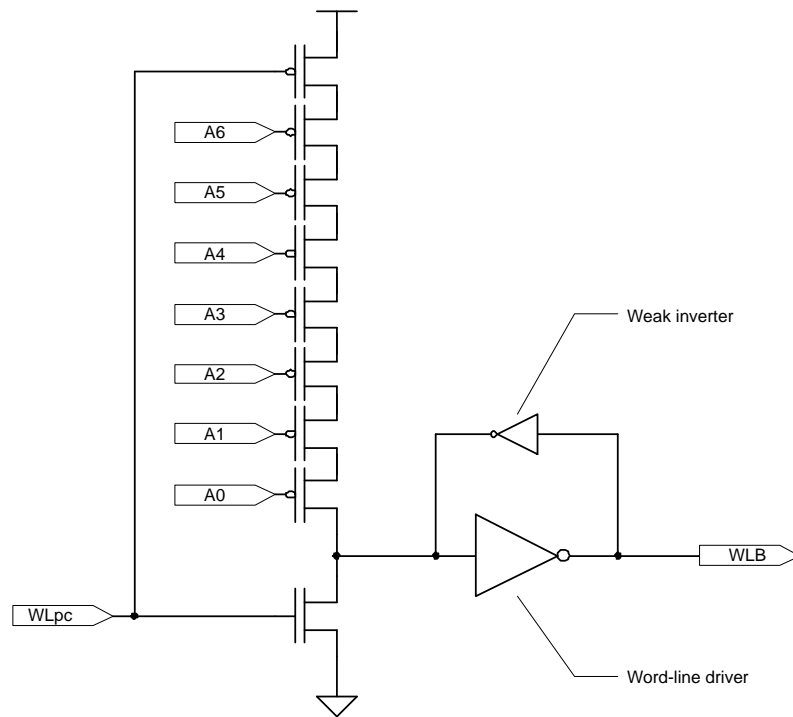


Figure 4.8: Word-line Decoder basic cell schematic.

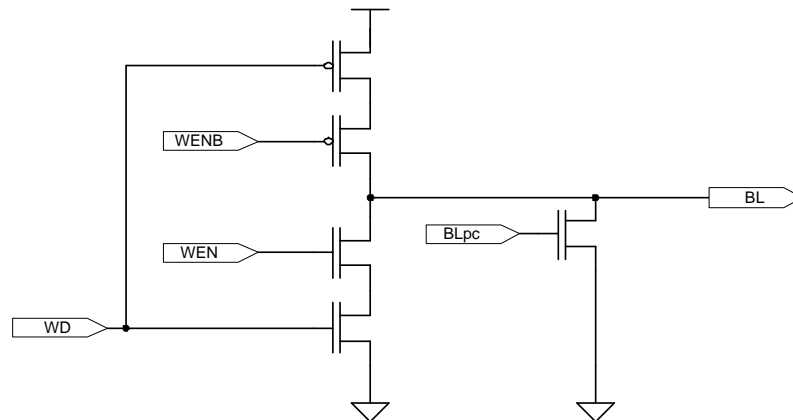


Figure 4.9: Bit-line Driver basic cell schematic.

The dummy bit-lines are driven by exactly the same cell, which nevertheless receives an hardwired value as input: high for `BLdummy` and low

for BLBdummy. Again, the cell has to be of the same horizontal size as the memory cell, and it follows that the vertical size is therefore big: about 25  $\mu\text{m}$ .

### 4.3.3 The Read Logic

The Read Logic is formed by simple tri-state inverters with a low threshold. This choice was done, instead of using sense-amplifiers, which draw a large amount of power, since the speed requirements for the memory are not tight. The inverters are switched to tri-state mode by the REN signal. The same considerations made for the Bit-line Driver are valid here: only one of the two blocks of the memory has to be read per time, and since the output bus is shared, the two blocks' tri-state inverters receive a different REN signal in order not to create conflicts.

## 4.4 Self-timing technique

Changing the memory array size, would make the bit-lines and the word-lines to vary their load, length, capacitance and therefore their delay accordingly. This means that, since the SRAM is designed for being size-configurable, the timing technique used has to take into account the possible adjustments in memory size: a self-timing technique is therefore employed.

The dummy word-line and the dummy bit-lines do the job: they are of the same length of the other corresponding lines, and have their same load, thus they have also the same capacitance and delay.

The dummy word-line is always de-asserted when beginning a read/write operation. A Timing Logic block probes this line's logic level to get the word-lines delay. The dummy bit-lines have instead almost the same behavior of the other bit-lines, thus they are strongly driven by the top buffer when writing, while weakly driven by a single cell when reading. The only exception is that the data stored in the dummy bit-lines will always be the same: BL to '1' and BLB to '0'.

The memory cells along the dummy word-line and the dummy bit-line differ from the normal ones. The dummy word-line cells are not connected to the bit-lines otherwise they would conflict with the data coming from the other cells. The dummy bit-line cells have instead high-impedance memory nodes, thus they don't interact with the dummy bit-lines when selected. Eventually, the common cell shared by the dummy word-line and the dummy bit-lines is hardwired at a specific stored value, previously mentioned, thus the same value will be always read from that cell. Figures 4.10, 4.11 and 4.12 show the three possible dummy memory cells.

*Dummy lines*

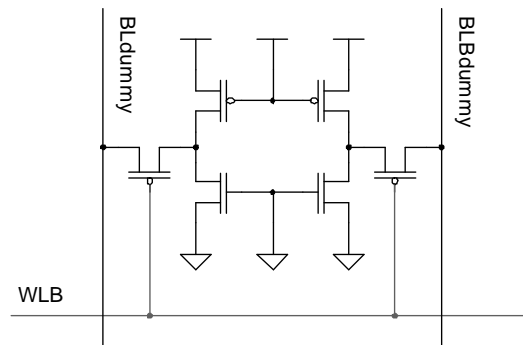


Figure 4.10: SRAM dummy bit-line cell.

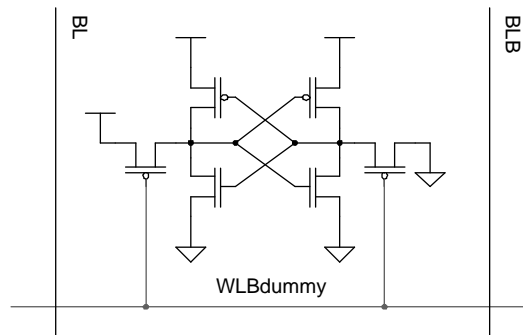


Figure 4.11: SRAM dummy word-line cell.

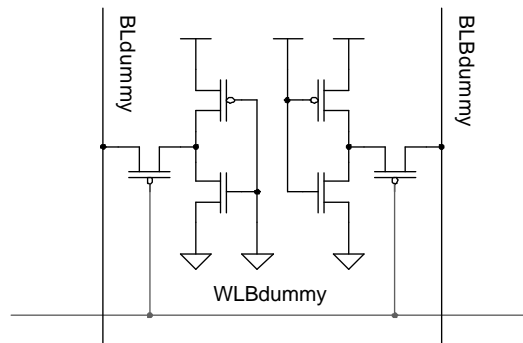


Figure 4.12: SRAM dummy cell at dummy bit-line/word-line intersection.

The cell 6-transistor structure is kept even in these dummy cells in order to have the maximum similarity with the normal memory cell geometry: the dummy lines have to match the normal lines.

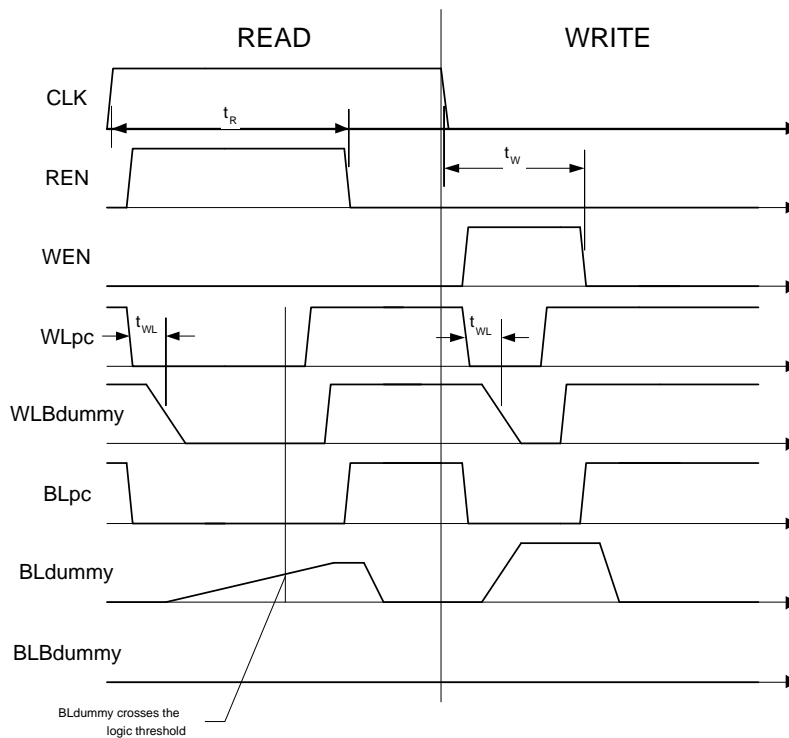


Figure 4.13: SRAM timing diagram showing both read and write operation.

The Timing Logic receives the dummy lines as well as the external clock and the R and W signals, which are respectively the read and write requests. Each read operation is initiated by the rising edge of the clock, while each write operation starts on the falling edge.

*Read/write timing*

The timing block decides when to pre-charge the word-lines through the WLpc output and when pre-discharge the bit-lines through the BLpc output. Moreover it enables (or disables) the writing and reading logic respectively with the two signals WEN and REN. Figure 4.13 shows the SRAM timing diagram.

Whenever a read operation begins, the REN read enable signal is raised and the Word-line Decoder de-asserts the selected word-line. When this reaches the low logic level the memory cells start driving the bit-lines. The cells are weak, thus bringing one of the two connected bit-lines to the high value takes some time, in the order of 1 ns. The Timing Logic checks the dummy bit-lines to know this delay, and when one of them reaches the logic threshold the read operation terminates. The REN is then lowered and a Data Output Latch, controlled by this signal, keeps the data on the output for the rest of the clock cycle. Immediately after the read ends, the word-lines are

restored to their pre-charge value and only after that, in order not to flip the memory content, also the bit-line are discharged.

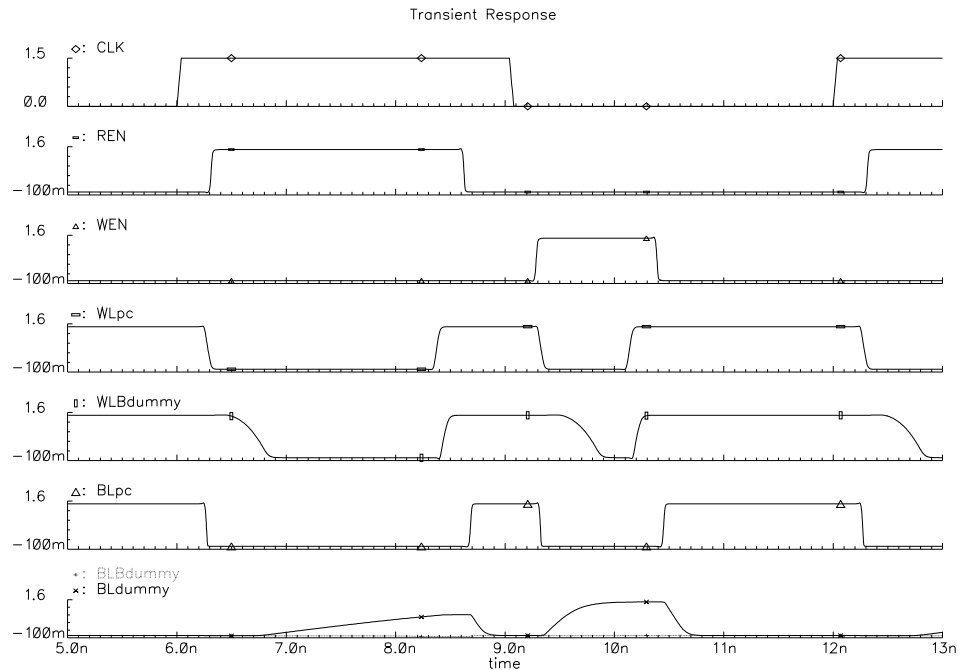


Figure 4.14: SPICE transient analysis on the SRAM in typical operating conditions. Parasitic capacitances are included. Voltages are expressed in V, while time is in ns.

A write operation starts instead with the **WEN** write enable signal going high: this allows the **Bit-line Driver** to prepare the right logic values on the bit-lines, meanwhile the **Word-line Decoder** de-asserts the selected word-line. Writing to the memory is faster than reading, since the driver is stronger than the cells, thus the **Timing Logic** terminates the write operation as soon as the dummy word-line reaches the zero logic level. The word-lines and bit-lines are then pre-charged, but their delay leaves enough time to complete the writing task correctly.

Figure 4.14 shows a SPICE transient analysis done on the whole SRAM with back-annotated parasitics and in typical conditions. Table 4.1 resumes the SRAM performance in 3 operating cases.

The **Timing Logic** is implemented using the 0.13  $\mu\text{m}$  library standard cells, placing them manually in order to minimize the area occupancy. The block lays outside of the core, among the input/output blocks.



Case	T [°C]	V <sub>DD</sub> [V]	Process	Read acc. time [ns]	Write acc. time [ns]	Max. clock freq. [MHz]
Best	-25	1.65	best	1.99	1.04	251
Typical	25	1.50	nominal	3.20	1.68	156
Worst	125	1.45	worst	3.56	1.87	140

Table 4.1: Results of several SPICE simulation on the SRAM. Read and write access times are the interval needed to complete the whole operation, measured from the clock edge to the bit-line pre-charge signal rise. The maximum operating frequency is referred to a 50% duty cycle clock. Process variation involve a few physical quantities like mobilities, saturation currents, etc. that are not reported here and are specified as recommended in the technology design manual.

## 4.5 Input/output blocks

A set of input/output blocks is needed for the memory operation. They are all implemented with standard cells placed manually next to the other and manually routed. Figure 4.15 illustrates the SRAM main blocks and their connections.

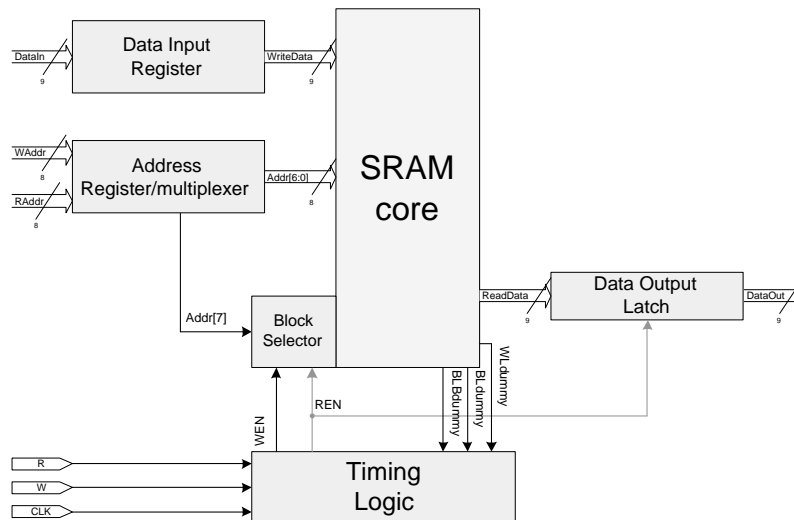


Figure 4.15: SRAM top-level view.

First of all, since the memory is dual-ported, it has two address buses, one for writing and one for reading. These two buses have to be multiplexed

*The Address register/multiplexer*

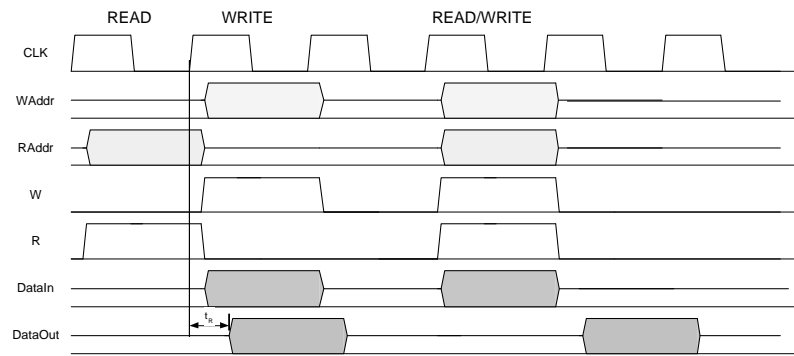


Figure 4.16: SRAM external timing.

on the internal address bus which brings the signals to the **Word-line Decoder** and the **Block Selector**. On top of that, the addresses have to be registered, since the behavior of the SRAM has to be synchronous from the outside point of view. The **Address Register/multiplexer** takes care of these tasks, sampling the addresses on the rising clock edge and putting the read address on the internal **Addr** bus when the clock is high, while the write address when the clock is low. Figure 4.16 describes the memory module's external timing.

#### *The Data Input Register*

The same considerations are valid for the data input, thus another block, the **Data Input Register** samples the data to be written on the clock rising edge.

#### *The Data Output Latch*

The data output comes from an asynchronous source and has to be latched when the read operation terminates: this is done by the **Data Output Latch**, which is enabled by the **REN** signal.

#### *The Block Selector*

Eventually, the **Block Selector** generates separate read and write enable signals (**REN** and **WEN**) for the two 9-bit blocks of the memory. This is done checking the most significative address bit.

## 4.6 Final layout

All the standard cells are placed in 4 vertical rows, sideways on the right in respect to the core. The core is surrounded by two  $5 \mu\text{m}$  thick power/ground vertical stripe pairs routed in the second metal layer, and two  $7.5 \mu\text{m}$  horizontal stripe pairs routed in the first metal layer. This guarantees a sufficient power distribution in the memory array. The final layout of the SRAM module is illustrated in Figure 4.17. The module measures  $129 \times 553 \mu\text{m}^2$ . Only

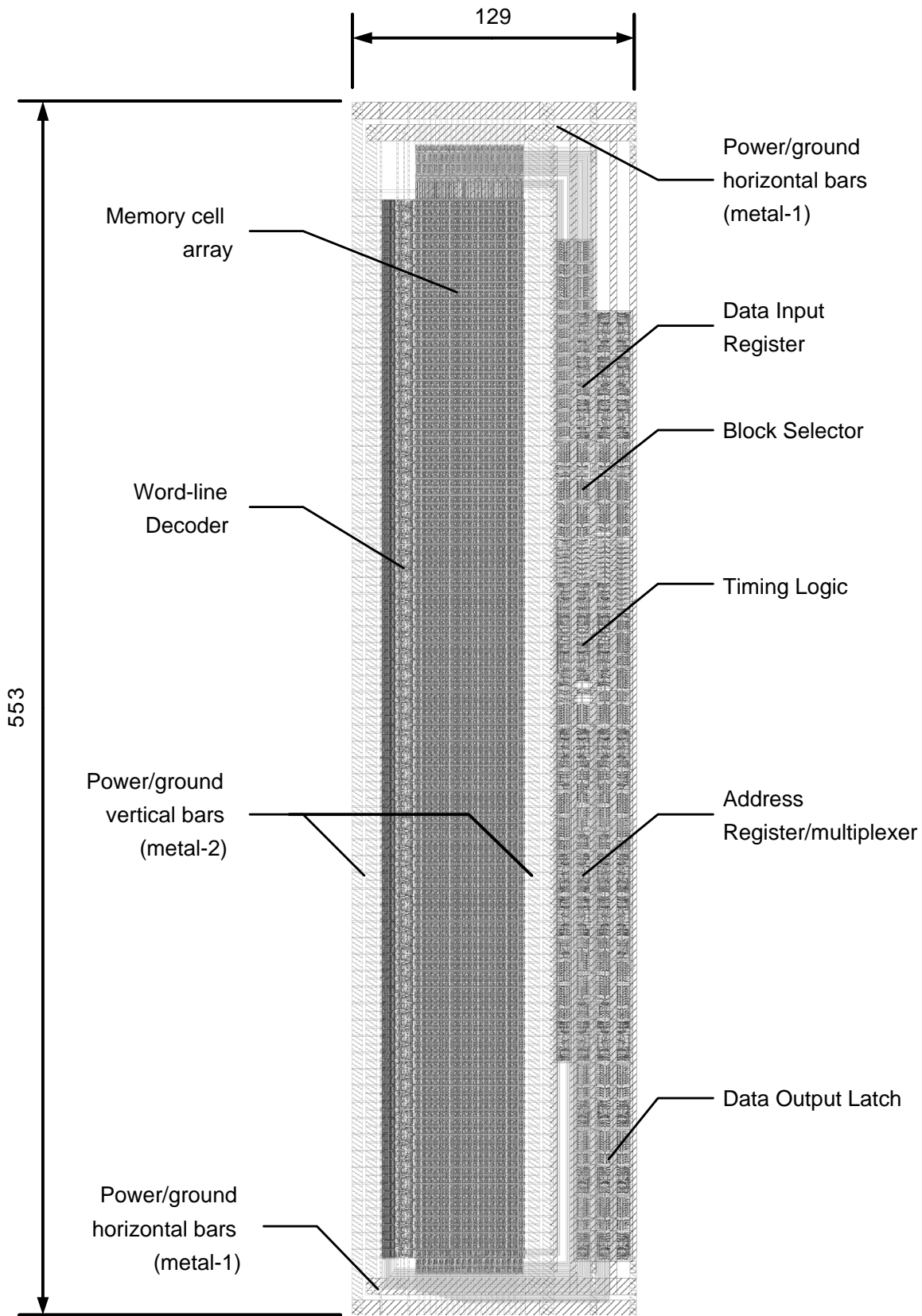


Figure 4.17: SRAM module layout. Measures are expressed in microns.

3 metal layers are used, saving precious resources for the ASICs which might include the block and can therefore route over the memory from the fourth metal upwards.

The final chip is by far pad-limited, but the choice of the memory size was based on the number of pads, which was fixed by the small area reserved for the chip on the technology testing wafer:  $2 \times 2 \text{ mm}^2$ . The actual chip size is  $1.84 \times 1.96 \text{ mm}^2$ , that fits exactly 46 pads. Two pairs of power supply for the module and two for the peripheral pads are included. A bigger SRAM module would have required more pads and therefore a bigger chip. Figure 4.18 shows the chip layout.

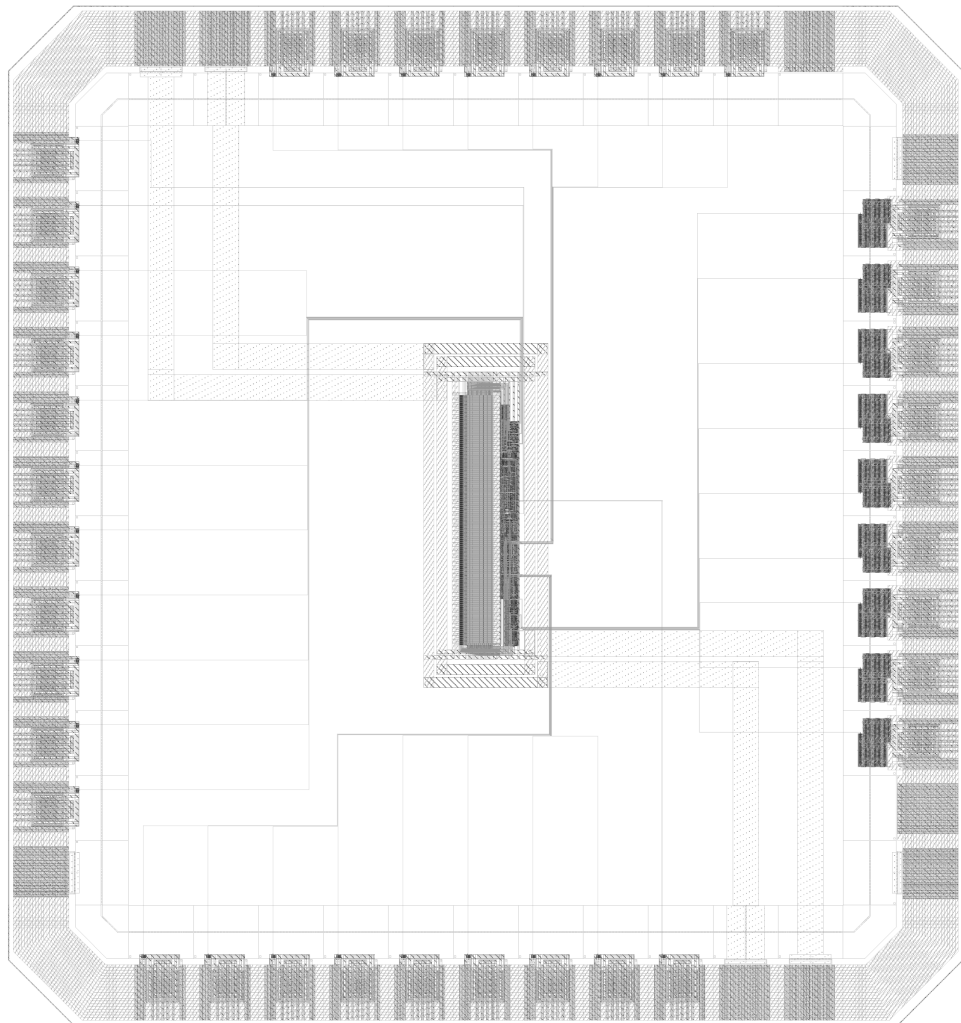


Figure 4.18: SRAM chip layout.

## 4.7 Future development and improvement

Future applications for the 0.13  $\mu\text{m}$  radiation tolerant SRAM might need a bigger memory size. Following the same low-power techniques used for [20], replicas of the memory array block used in the present design can be added to both the sides of the Word-line Decoder, and making use of word-line buffers for each additive block. The word-line distribution can then be done in two steps: the Word-line Decoder outputs the global word-line, while each buffer generates the local word-line. In order to have low-power consumption, only the selected block's buffer will be enabled, leaving the rest of the local word-lines pre-charged. The Read Logic could also be placed vertically centered in the memory, while putting the Bit-line Driver on both the top and bottom sides. The resulting core is represented in Figure 4.19.

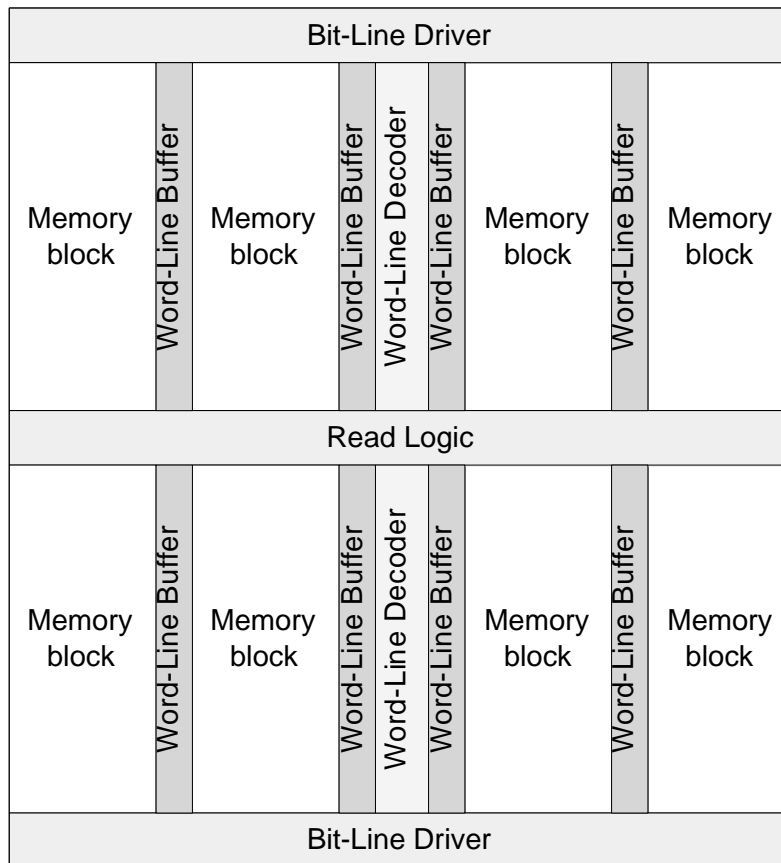


Figure 4.19: SRAM core for bigger memory sizes.



## Appendix A

# The Delay Locked Loop

A Delay Locked Loop (DLL) is able to add delay to an input signal by a certain fraction of the clock period, and is based on two (or more) identical delay chains. Figure A.1 shows a DLL basic block diagram.

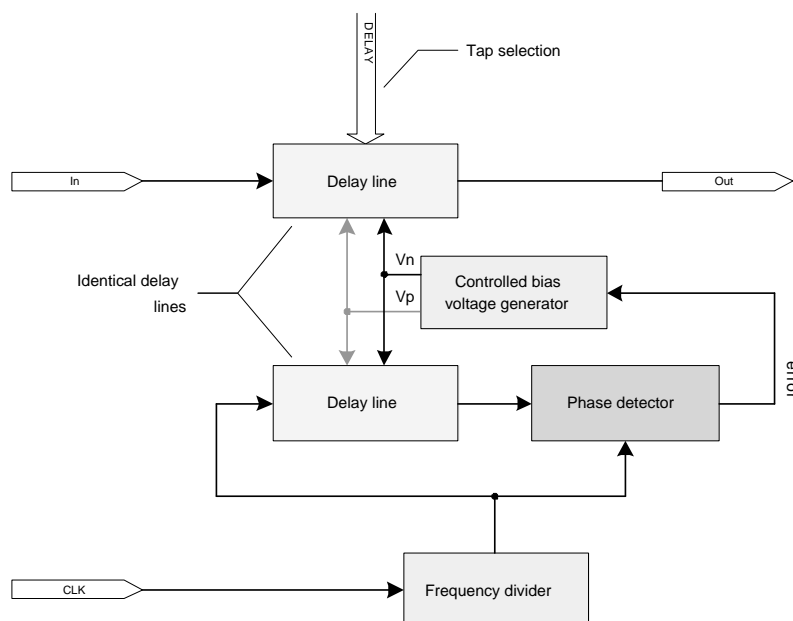


Figure A.1: Delay Locked Loop basic block diagram.

One of the delay chains serves as reference, while the other effectively delays the input signal. Both chains need a bias voltage to operate, which comes from a bias voltage generator. The reference chain is part of a feedback loop which involves also the bias voltage generator controlled by a phase detector: in this way the delay of the reference chain will be kept equal as much as possible to an internal reference clock period, which is a multiplied version of the clock input. In practice, the phase detector will

signal the differences between the reference chain's delay and the internal clock period, making the bias voltage generator adjust its output bias to the chain. Figure A.2 illustrates a delay chain.

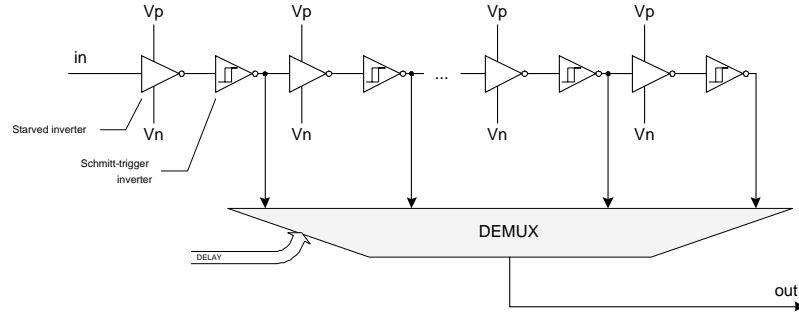


Figure A.2: Delay chain.

Every delay chain is composed by a repetition of the same delay element, as many times as the desired clock period division. Through a de-multiplexer it is possible to select among the taps in the chain to obtain the proper delay. The reference chain always gives the maximum delay as output. Each one of the delay elements is composed by a *starved inverter* and a Schmitt-trigger inverter. The first one is a special inverter which has a slew rate controlled by the bias voltage given: Figure A.3 shows a starved inverter. Two transistors

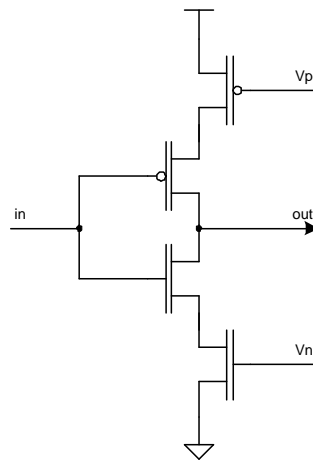


Figure A.3: Starved inverter.

on the current path limit the current amount, according to their gate bias. In this way it is possible to regulate the inverter's delay. Undefined states, output by the starved inverter are resolved by the Schmitt-trigger inverter.



## Appendix B

# A triplicated parameterized CRC generator

As an example of triplicated state machine HDL description here is reported the Verilog code used for the CRC Generator in the Kchip (see Section 3.2.15). Three modules are defined: an open state machine, which has state input and state output, plus a higher level block which instantiates three times the leaf cell and adds a majority voting logic, defined in the third module.

```
// Author:      BONACINI Sandro, CERN EP/MIC
//
// CRC Generator/Checker
//
// CRC is calculated right-shifting, the polynomial must therefore
// be reversed.
// The output crc is directly the crc register value. There is no
// inversion in between.
// Default parameters produce a CRC16-CCITT output.
// When checking, correct reception is signaled by a zero in the crc
// output.
//
// Parameters:
//
// CRC_WIDTH      bit width of the crc output;
//
// DATA_WIDTH    bit width of the data input (must be at least 2);
//
// INIT_VAL      initialization value of the crc register,
//               suggested values are all-zeros and all-ones;
//
// POLY          Polynomial (remember to reverse it)
//               i.e. CCITT 1021h has POLY = 'h8408.
//
////////////////////////////////////////////////////////////////
```

```

`timescale 1ns/1ns
`undef delay
`define delay 5
module crc_iostate (d, init, reset_b, clk, d_valid, crc_out, crc_in);
    // synopsys template

    parameter CRC_WIDTH = 16;
    parameter DATA_WIDTH= 16;
    parameter INIT_VAL   = 'hffff;
    parameter POLY       = 'h8408;

    // I/Os

    input  [(DATA_WIDTH-1): 0] d;
    input                               init;
    input                               d_valid;
    input                               clk;
    input                               reset_b;
    input  [(CRC_WIDTH-1) : 0] crc_in;

    output [(CRC_WIDTH-1) : 0] crc_out;

    // Output regs

    reg    [(CRC_WIDTH-1) : 0] crc_out;

    // Internal wires & regs

    reg    [(CRC_WIDTH-1) : 0] next_crc;

    // Always statements

    always @(d or crc_in) begin
        next_crc = crc_calc(crc_in, d);
    end

    // synopsys async_set_reset "reset_b"
    // synopsys sync_set_reset "init"
    always @ (posedge clk or negedge reset_b)
    begin
        if (~reset_b) begin
            crc_out <= #'delay 0;
        end
    end

```



```

`define delay 5

module crc_tri (d, init, reset_b, clk, d_valid, crc, SEU_err);

    parameter CRC_WIDTH = 16;
    parameter DATA_WIDTH= 16;
    parameter INIT_VAL  = 'hffff;
    parameter POLY      = 'h8408;

    // I/Os

    input  [(DATA_WIDTH-1): 0] d;
    input                               init;
    input                               d_valid;
    input                               clk;
    input                               reset_b;

    output [(CRC_WIDTH-1) : 0] crc;
    output                               SEU_err;

    // Internal wires

    wire  [(CRC_WIDTH-1) : 0] crc_1, crc_2, crc_3;

    // Module instantiations

    crc_iostate #(CRC_WIDTH, DATA_WIDTH, INIT_VAL, POLY) sm1 (
        .d          (d),
        .init       (init),
        .reset_b    (reset_b),
        .clk        (clk),
        .d_valid    (d_valid),
        .crc_out    (crc_1),
        .crc_in     (crc)
    );

    crc_iostate #(CRC_WIDTH, DATA_WIDTH, INIT_VAL, POLY) sm2 (
        .d          (d),
        .init       (init),
        .reset_b    (reset_b),
        .clk        (clk),
        .d_valid    (d_valid),
        .crc_out    (crc_2),
        .crc_in     (crc_1)
    );

```

```

    );

    crc_iostate #(CRC_WIDTH, DATA_WIDTH, INIT_VAL, POLY) sm3 (
        .d            (d),
        .init         (init),
        .reset_b      (reset_b),
        .clk          (clk),
        .d_valid      (d_valid),
        .crc_out      (crc_3),
        .crc_in       (crc_2)
    );

    majority_voter #(CRC_WIDTH) mv (
        .in1(crc_1),
        .in2(crc_2),
        .in3(crc_3),
        .out(crc),
        .err(SEU_err)
    );

endmodule

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

module majority_voter (in1, in2, in3, out, err);
    // synopsys template

    parameter WIDTH = 1;

    input    [(WIDTH-1):0]    in1, in2, in3;

    output   [(WIDTH-1):0]    out;
    output                                     err;

    reg     [(WIDTH-1):0]    out;
    reg     [(WIDTH-1):0]    err;

    always @(in1 or in2 or in3) begin
        err = 0;
        out = vote (in1,in2,in3);
    end

    function vote_atom;

```

```
input in1,in2,in3;
begin
if (in1 == in2) begin

    vote_atom = in1;

    if (in2 != in3) err = 1;

end
else begin
    vote_atom = in3;
    err = 1;
end
end
endfunction

function [(WIDTH-1):0] vote;
input [(WIDTH-1):0] in1, in2, in3;
integer i;
begin
for (i=0; i<WIDTH; i=i+1)
    vote[i] = vote_atom( in1[i], in2[i], in3[i] );
end
endfunction

endmodule
```

# Bibliography

- [1] G. Anelli. *Design and characterization of radiation tolerant integrated circuits in deep submicron CMOS technologies for the LHC experiments*. PhD thesis, Insitut National Polytechnique de Grenoble, France, December 2000.
- [2] G. Anelli et al. Total dose behavior of submicron and deep submicron CMOS technologies. In *3th Workshop on electronics for LHC experiments*, London, September 1997.
- [3] H.E.Jr. Boesch et al. Saturation of threshold voltage shift in MOSFETs at high total dose. *IEEE Transactions on Nuclear Science*, 33(6):1191–1197, December 1986.
- [4] H.E.Jr. Boesch and F.B. McLean. Hole transport and trapping in field oxides. *IEEE Transactions on Nuclear Science*, 32(6), December 1985.
- [5] D. Braunig. Ionization and Displacement. In *Notes of the Short Course of the 2nd European Conference on Radiation and its Effects on Components and Systems*, number 2 in , Saint-Malo, France, September 1993.
- [6] J. Christiansen. Testing LHC electronics. In *5th Workshop on electronics for LHC experiments*, Snowmass, Colorado, September 1999.
- [7] CMS. The Compact Muon Solenoid. Technical proposal, CERN, December 1994. CERN/LHCC/94-38.
- [8] CMS. The Electromagnetic Calorimeter Project. Technical Design Report 4, CERN/CMS, December 1997. CERN/LHCC/97-33.
- [9] A.G.F. Dingwall and R.E. Stricker. C<sup>2</sup>L: A new high-speed high-density bulk CMOS technology. *IEEE Journal of Solid-State Circuits*, 12, August 1977.
- [10] F. Faccio, G. Anelli, et al. Total dose and Single Event Effects (SEE) in a 0.25  $\mu\text{m}$  CMOS technology. In *4th Workshop on electronics for LHC experiments*, Roma, September 1998. Università di Roma “La Sapienza”.

- [11] F. Faccio, K. Kloukinas, G. Magazzù, and A. Marchioro. SEU effects in registers and in a Dual-Ported Static RAM designed in a 0.25 micron CMOS technology for applications in the LHC. In *5th Workshop on electronics for LHC experiments*, Snowmass, Colorado, September 1999.
- [12] A. Giraldo. *Evaluation of Deep Submicron Technologies with Radiation Tolerant Layout for Electronics in the LHC Environments*. PhD thesis, University of Padova, Italy, December 1998. URL: <http://wwwcdf.pd.infn.it/cdf/sirad/giraldo/tesigiraldo.html>.
- [13] M. Huhtinen. Studies of neutron moderator configurations around the CMS inner tracker and ECAL. Technical note 057, CERN/CMS, 1996.
- [14] M. Huhtinen. Method for estimating dose rates from induced radioactivity in complicated hadron accelerator geometry. Divisional report, CERN/TIS, 1997.
- [15] P. Jarron, G. Anelli, T. Calin, et al. Deep submicron CMOS technologies for the LHC experiments. *Nuclear Physics B (Proceedings Supplement)*, 78:625–634, August 1999. issues 1–3.
- [16] P. Jarron, G. Anelli, et al. Study of the radiation tolerance of IC's for LHC. RD49 status report 2, CERN/MIC, March 1999. CERN/LHCC/99-8.
- [17] P. Jarron, G. Anelli, et al. Study of the radiation tolerance of IC's for LHC. RD49 status report 3, CERN/MIC, January 2000. CERN/LHCC/2000-03.
- [18] K. Kloukinas. CMS preshower front-end readout and control system. Draft, CERN/CME, 2001. v0.2.
- [19] K. Kloukinas, F. Faccio, A. Marchioro, and P. Moreira. Development of a radiation tolerant 2.0 V standard cell library using a commercial deep submicron CMOS technology for the LHC experiments. In *4th Workshop on electronics for LHC experiments*, Roma, September 1998. Università di Roma “La Sapienza”.
- [20] K. Kloukinas, G. Magazzù, and A. Marchioro. A Configurable Radiation Tolerant Dual-Ported Static RAM macro, designed in a 0.25  $\mu\text{m}$  CMOS technology for applications in the LHC environment. In *8th Workshop on electronics for LHC experiments*, Colmar, France, September 2002.
- [21] J.B. Kuo and J.H. Luo. *Low Voltage CMOS VLSI Circuits*. Wiley Interscience, J. Wiley and Sons, 1999.



- [22] F. Lemeilleur et al. Study of characteristics of silicon detectors irradiated with 24 GeV/c protons between  $-20$  °C and  $20$  °C. Divisional report, CERN/ECP, 1994. CERN-ECP/94-8.
- [23] A. Marchioro. Deep submicron technologies for HEP. In *4th Workshop on electronics for LHC experiments*, Roma, September 1998. Università di Roma “La Sapienza”.
- [24] F.B. McLean, H.E.Jr. Boesch, and T.R. Oldham. *Ionizing Radiation Effects in MOS Devices & Circuits*, chapter Electron-Hole generation, transport and trapping in SiO<sub>2</sub>. J.Wiley & Sons, New York, 1989.
- [25] P.J. McWhorter, S.L. Miller, and W.M. Miller. Modeling the anneal of radiation-induced trapped holes in a varying thermal environment. *IEEE Transactions on Nuclear Science*, 37(6):1682–1688, December 1990.
- [26] P. Moreira, T. Toifl, A. Kluge, G. Cervelli, A. Marchioro, and J. Christiansen. The GOL Reference Manual. Preliminary v1.1, CERN/EP/MIC, January 2002.
- [27] W. Snoeys, G. Anelli, M. Campbell, et al. Integrated circuits for particle physics experiments. *IEEE Journal of Solid-State Circuits*, 35(12), December 2000.
- [28] R. Thomas and G.R. Stevenson. Radiological safety aspects of the operation of proton accelerators. Technical Report Series 283, IAEA, 1988.
- [29] P.S. Winokur et al. *Ionizing Radiation Effects in MOS Devices & Circuits*, chapter Radiation-Induced Interface Traps. J.Wiley & Sons, New York, 1989.
- [30] C. Yen, R. Walker, P. Petruno, C. Stout, B. Lai, and W. McFarland. G-Link: A chipset for Gigabit-Rate Data Communication. *Hewlett-Packard Journal*, October 1992.