EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
ORGANISATION EUROPEENNE POUR LA RECHERCHE NUCLEAIRE

CERN - PS DIVISION

PS/BD/ Note 98-09 (Tech.)

BURST AND RF TRIGGER GENERATOR MODULES (VMOD-BURST AND VMOD-RFTRIG)

J.L. Gonzalez, J.T. Pons

ABSTRACT

This note describes the Burst and RF Trigger Generator piggyback modules. The Burst Generator produces a pre-programmed number of pulses, equally spaced, after reception of an external trigger. The RF Trigger Generator synchronises the burst pulses to the external clock signals and allows setting a delay and the pulse-width. Two pairs of these modules can be installed on a VMOD-IO VME module. Although originally they are realised for the PS and PSB Q-Measurement systems, these modules are useful in any application requiring bursts or synchronised signals.

Geneva, Switzerland
28 July 1998

1. Introduction

The Burst and the RF Trigger Generators are independent piggyback [1] modules that can be installed on a JANZ VMOD-IO VME module [2]. The Burst Generator produces bursts of pulses, after reception of an external trigger. The number of pulses and their spacing are programmable. The RF Trigger Generator synchronises these pulses to external clock signals and allows setting a delay and the pulse-width.

The core of the design takes the form of a list of logic equations, implemented in the Altera EPM7128E-84 PLD [3]. Both modules use the same pin assignments, in order to fit the same printed circuit board design.

Although originally they are realised for the Q-Measurement system of the PS and PSB accelerators, these piggyback modules can be used in any system requiring bursts or synchronised signals. Two pairs of these modules can be installed on a VMOD-IO module, one for the horizontal plane and another for the vertical plane.

2. The Burst Generator Module

The Burst Generator module (VMOD-BURST) produces a burst, after reception of an external trigger. This module has three inputs (*Trigger*, *Ext_Train* and *Reset*) and one output (*Burst_out*). These signals are active low. The number of output pulses and their spacing are user programmable, and they are synchronised to both an internal 2 MHz clock and an external/internal *Train* signal. The *Reset* input can be used to stop a burst initiated by a previous trigger. Figure 1 shows the block diagram of this module.

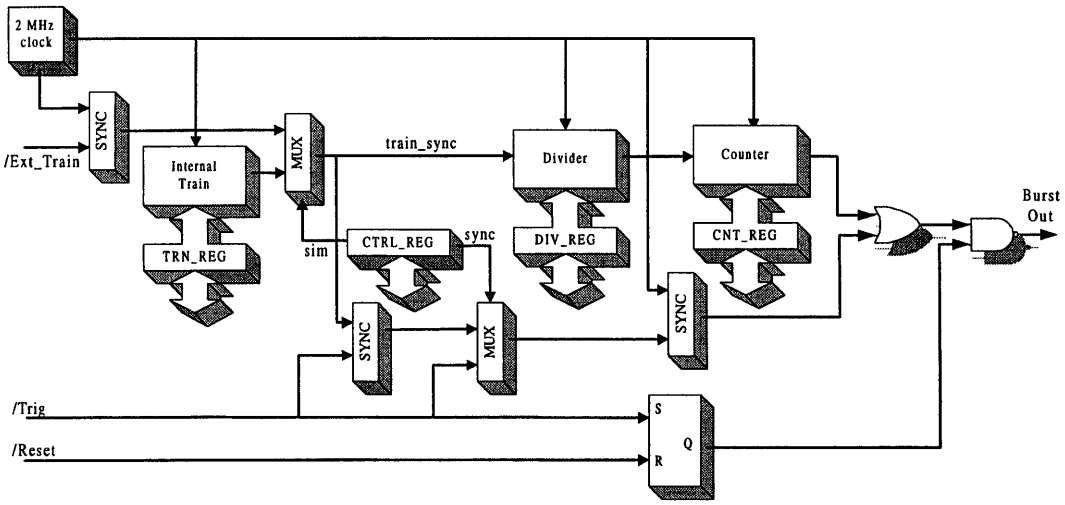


Figure 1: Block diagram of the Burst Generator module

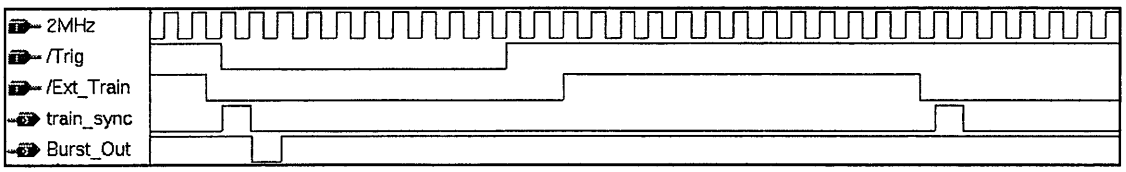


Figure 2: Synchronisation timing example (with 2 MHz and /Ext_Train)

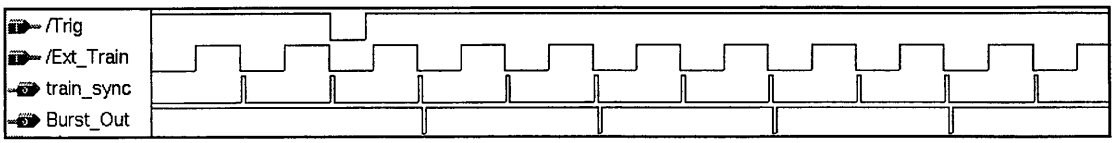
The example of Figure 2 shows the */Ext_Train* input synchronised with the *2 MHz* clock, which produces the synchronising *train_sync* signal. The first *Burst_Out* pulse is a special case, where the trigger signal is synchronised only with the *2 MHz* clock.

The VMOD-BURST module has a 2-bit read/write *Control Register* and three 12-bit read/write registers: TRN_REG, DIV_REG and CNT_REG. The *Train Register* (TRN_REG) is used to simulate the External Train, from the 2 MHz clock (maximum period duration is about 2 ms, due to counter size). The *Divider Register* (DIV_REG) contains the divider value for the burst train. Pulse outputs are produced when the divider settings are greater than or equal to one. The *Counter Register* (CNT_REG) establishes the number of pulses that must be generated after a trigger. It must be set to M to obtain M+1 pulses (when M = 0, only one pulse is created). The *Control Register* bits have the following functions:

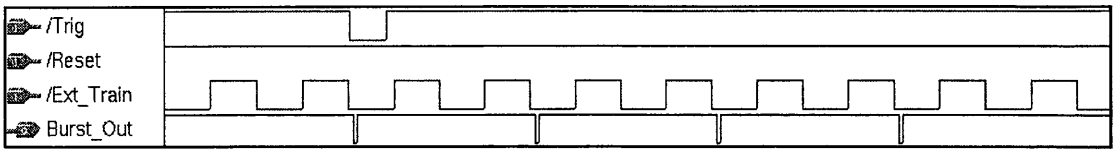
Bit 1	Bit 0
sync	sim

Bit 0 (*sim*) selects an external or an internal source for the *train_sync* signal: *sim* = 0 activates the external train, and *sim* = 1 produces a simulated clock. The period of this clock is (K+1)×500 ns, where K is the contents of register TRN_REG.

Bit 1 (*sync*) disables the synchronisation of the *Trigger* signal to the *train_sync*. When *sync* = 0, the *Trigger* signal is synchronised to both the *train_sync* and the *2 MHz* clock. If *sync* = 1, it is synchronised only to the *2 MHz* clock. Figure 3 presents both cases. In example (a), the double synchronisation introduces a maximum delay of one *train_sync* period plus 1 μs, while in (b), the trigger pulse gets out without synchronising to the *train_sync* signal (the delay between trigger input and first output pulse is 1 μs). The pulse-width corresponds to the internal clock period (500 ns).



(a) *sync* = 0



(b) *sync* = 1

Figure 3: Output signal, as a function of the control register *sync* bit

3. The RF Trigger Generator Module

The RF Trigger Generator module (VMOD-RFTRIG) allows setting the delay and the pulse-width of the burst signal. Block diagram, in Figure 4, shows that the module has three inputs (*Burst_in*, f_{rev} and f_{RF}) and one output (*RF_Out*). These signals are active low. The output pulses are synchronised to both f_{rev} and f_{RF}. Moreover, these external clocks can be generated internally.

The VMOD-RFTRIG module has a read/write *Control Register* (1-bit) and three 8-bit read/write registers: DEL_REG, WID_REG and FREV_REG. The *Delay Register* (DEL_REG), which contains the delay of the output signal, and the *Width Register* (WID_REG), which controls the pulse width, are both expressed as a number of f_{RF} periods. When WID_REG = 0, no output is produced. The *Control Register* (CTRL_REG) selects internal or external sources. When set to zero, the *Burst_IN* input is synchronised to the external signals f_{rev} and f_{RF}; otherwise the internal clock generates both. In that case, f_{RF} is a 10 MHz clock and f_{rev} corresponds to f_{RF} divided by M, where M is the value stored in the f_{rev} *Register* (FREV_REG).

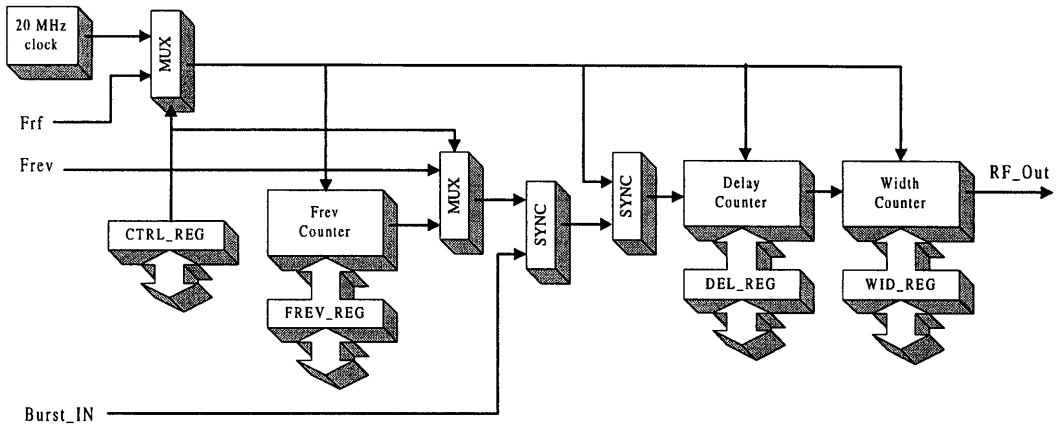


Figure 4: Block diagram of the RF Trigger Generator module

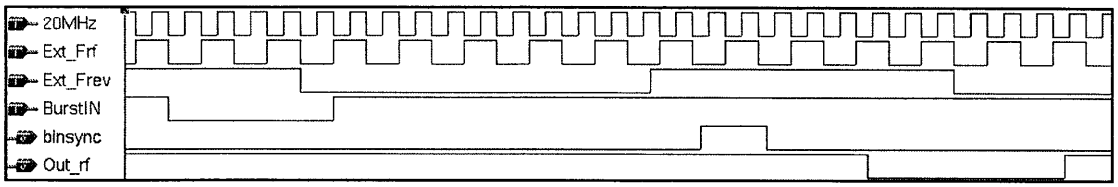


Figure 5: Timing diagram of the RF Trigger Generator Module

On Figure 5, the *Delay Register* contains '2' and the *Width Register* is set to '3'. Therefore, the output is delayed by 2 f_{RF} periods and the pulse width is 3 f_{RF} periods. The *Burst_IN* pulse is synchronised first to f_{rev} and then to f_{RF}.

4. Implementation

The VMOD-BURST and the VMOD-RFTRIG are independent piggyback modules. They must be installed on a VMOD-IO motherboard, which can hold up to 4 plug-in modules with MODULbus interfaces. Each MODULbus (piggyback) socket has a 512 byte address space and can be selected either as byte-device or as word-device. The addresses of the MODULbus sockets are:

MODULbus	Hex Address range
socket 0	0 - 1FF
socket 1	200 - 3FF
socket 2	400 - 5FF
socket 3	600 - 7FF

Both modules use an Altera EPM7128E 84-pin PLD, with the same pin assignments, in order to fit the same printed circuit board. Some spare pins have been devoted to drive internal signals, for test purposes.

The VME interface of both modules consists of four registers. Their memory maps are shown on the following tables:

VMOD-BURST		
Register	**Address (r/w)**	**No. Bits**
CTRL_REG	VME BASE + MODULbus BASE + 0	2
TRN_REG	VME BASE + MODULbus BASE + 2	12
DIV_REG	VME BASE + MODULbus BASE + 4	12
CNT_REG	VME BASE + MODULbus BASE + 6	12

VMOD-RFTRIG		
Register	**Address (r/w)**	**No. Bits**
CTRL_REG	VME BASE + MODULbus BASE + 0	1
DEL_REG	VME BASE + MODULbus BASE + 2	8
WID_REG	VME BASE + MODULbus BASE + 4	8
FREV_REG	VME BASE + MODULbus BASE + 6	8

6. Application to the Q-Measurement system for PS and PSB accelerators

A VMOD-BURST and a VMOD-RFTRIG are already installed in the PS FFT Q-measurement system. These modules have replaced the old CAMAC preset-counters, which generated the kicker timing. Thus, the CAMAC crate and the VME Serial CAMAC driver have been removed. The real-time program of the Q-measurement system [4] has been updated to suppress the related CAMAC functions and add control of this new hardware.

These piggyback modules are more powerful than the CAMAC system. Indeed, all the features of the CAMAC are implemented in the VMOD-BURST module. The second module, the VMOD-RFTRIG, can be used to select one bunch inside a revolution period. The DEL_REG register allows adjusting the output pulse delay, within the limit of 256 f_{RF} cycles. It is also possible to set up the pulse-width to excite the desired number of bunches. In addition, the f_{rev} and f_{RF} signals can be simulated. These features were not realised in the old CAMAC.

The *Ext_Train* signal in the PS accelerator (named *C_Train*) is a 1 KHz clock. For test purposes, it can be simulated by setting the TRN_REG register to 2000 in the VMOD-BURST module. The burst generator counts up to 4096 (CNT_REG is a 12-bit register), allowing for cycles as long as 4 seconds. Since the PS magnetic cycle can last up to 2.4 s, then 2400 counts are necessary to cover the whole cycle with the *C_Train*.

Note that, in the PS, the Q-Measurement trigger signal is already derived from the *C_Train*. Then, synchronising it again will result in an additional delay of a *C_Train* period. Therefore, in order to avoid that resynchronisation, the *sync* bit of the control register must be set to "1".

7. Software

This section gives the guidelines to program the VMOD-BURST and VMOD-RFTRIG in the DSC environment, i.e. using the PS library interface. A hardware module type has been created for each piggyback. The source program using the library interface must include the header file *<drviutil/ioconfiglib.h>*.

The *IocModulPointer()* function must be called, in order to get a hardware pointer to (part of) the space of the specified module (*type, lun*). The syntax is:

```
Extern int IocModulPointer (
       int  type,        /* type of module (see moduletypes.h)          */
       int  lun,         /* logical unit number of target               */
       int  part,        /* target part                                 */
       char **modulptr   /* receive the VME pointer in the target space */
       );
```

The functions *IocVMODBURST* and *IocVMODRFTRIG* give respectively the module type of the VMOD-BURST and VMOD-RFTRIG. If successful, these functions return 0, otherwise they return an error code (negative value).

Figure 6 shows a pseudo-code example.

```
#include <drvrutil/ioconfiglib.h>
...
typedef struct {
  unsigned short ctrl;
  unsigned short train;
  unsigned short div;
  unsigned short cnt;
} burst;

typedef struct {
  unsigned short ctrl;
  unsigned short delay;
  unsigned short width;
  unsigned short frev;
} rftrig;

burst  *bm;
rftrig *tm;
char   *pburst, *ptrig ;

if (IocModulPointer(IocVMODBURST, 0, 0, &pburst) < 0) {
    mess_err(ErrFATAL, "Unable to get VMOD-BURST pointer\n");
  ...
}

if (IocModulPointer(IocVMODRFTRIG, 0, 0, &ptrig) < 0) {
    mess_err(ErrFATAL, "Unable to get VMOD-RFTRIG pointer\n");
  ...
}

bm = (burst *) pburst;
tm = (rftrig *) ptrig;
```

Figure 6: Pseudo-code programming example

Two data structures have been created with the register structure of the modules. The pointer to the base address of the module, obtained by calling the *IocModulPointer()* function, is assigned to the data structure pointer. Figure 7 shows how to program the piggyback modules with the same functionality as the old CAMAC modules.

```
/* Burst Generator */
    bm->ctrl = 0x2;        /* External signal. No sync with C_Train */
    bm->div  = period;     /* C-Train divider                       */
    bm->cnt  = (number_of_pulses >=1) ? (number_of_pulses -1) : 0;

/* RF Trigger Generator */
    tm->ctrl  = 0x0;       /* External frf and frev       */
    tm->delay = 0x0;       /* No delay                    */
    tm->width = 0xa;       /* pulse-width: 10 frf periods */
```

Figure 7: Programming the piggyback modules to replace the old CAMAC

8. Acknowledgments

We are indebted to J. Belleman and A. Chapman-Hatchett for their kind advice and helpful comments.

References

[1] MODULbus specification manual, ver. 2.2n, MODULbus e.V., 1997.

[2] VMOD-IO Hardware manual, ver. 2.2, JANZ Computer AG, Im Dörener Feld 3, D-4790 Paderborn, Germany.

[3] Altera, Max+PlusII, Programmable Logic Development System Manuals.

[4] S. Johnston, *Real-Time Program for the PS FFT Q-Measurement*, CERN/PS/Note 94-2, January 1994.