Performance Analysis of the IXP1200 Network Processor

Rajesh Krishna Balan and Urs Hengartner

Introduction

- Network Processors are currently of great interest to many companies (Intel, IBM etc.)
- They integrate the best features of custom ASICs and general purpose processors
- The Intel IXP1200 was one of the first Network Processors available in the market.
- This project revolves around measuring the performance of the IXP1200.

IXP1200 Architecture

- StrongArm core processor with 6 programmable RISC based microengines
- StrongArm core is used for control plane functionality (routing table maintenance etc.) while microengines handle data plane functionality (packet classification, packet I/O etc.)
- Scratchpad RAM / SRAM / SDRAM
- Standard 32bit PCI bus and fast 64bit IX bus controlled by FBI unit

Microengines

- 6 microengines
  - Single pipelined in-order RISC processor with customized instruction set optimized for packet processing
  - Each microengine can run 4 separate threads with a different PC for each thread and hardware assisted super-fast context switching
- 256 registers per microengine
  - 128 32bit general purpose registers
  - 128 32bit transfer registers to access SDRAM and SRAM
- 32 bit ALU and shifter

Stage	Description
P0	Instruction lookup
P1	Instruction decode and formation of source register address
P2	Read of operands from source registers
P3	Perform ALU, shift, or compare operations
P4	Write result to destination register

Table 1: Work per pipeline stage.

Architecture

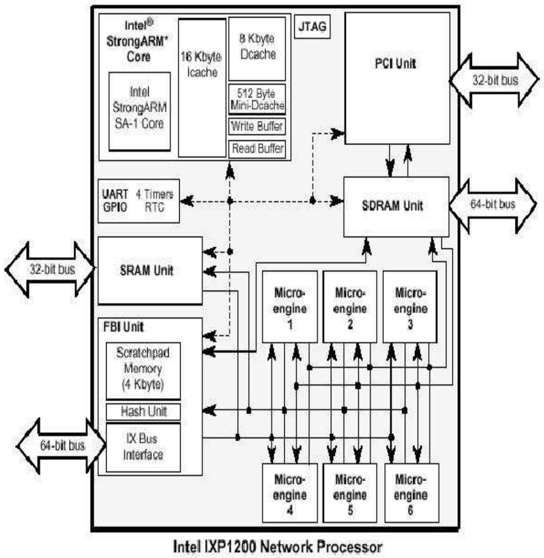


Figure 1: Architecture of the Intel IXP1200 network processor. (*Source:* Intel IXP1200 Network Processor Datasheet, page 11)

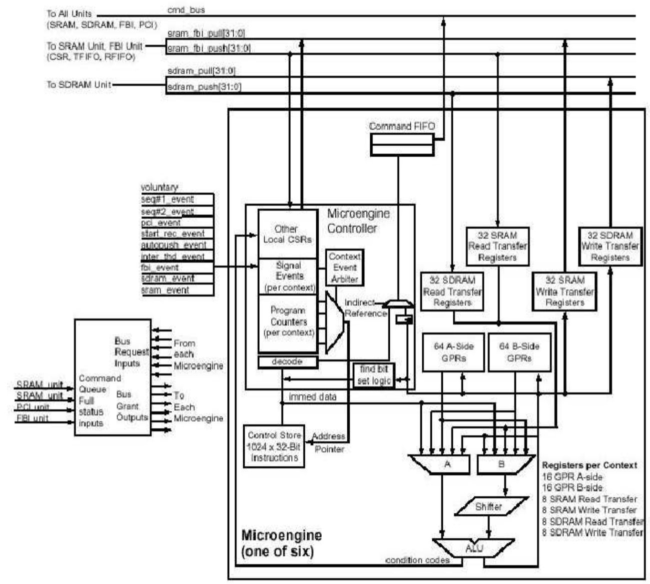


Figure 2: Architecture of a microengine. (*Source:* Intel IXP1200 Network Processor Hardware Reference Manual, page 4-2)

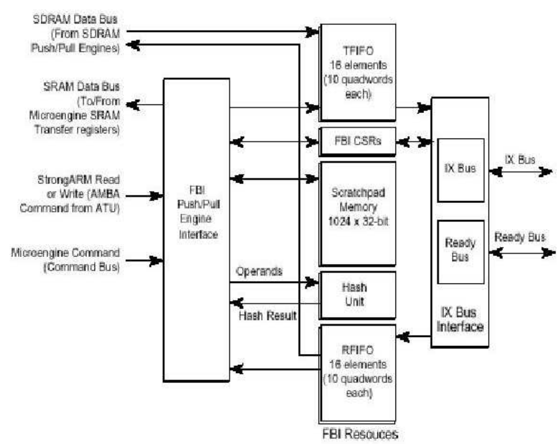
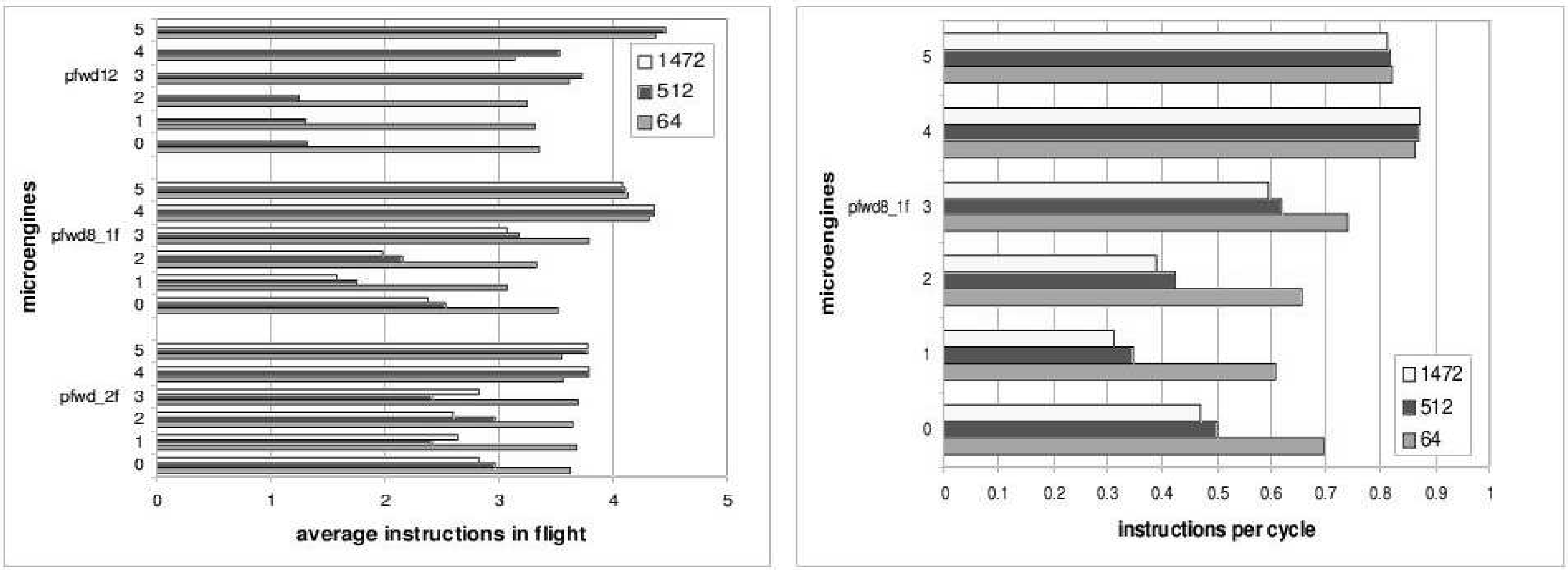


Figure 3: Architecture of the FBI unit. (*Source:* Intel IXP1200 Network Processor Hardware Reference Manual, page 6-2)

IPC / Instructions in Flight



(a) Average number of instructions in flight.

(b) Average IPC.

Figure 4: Instructions in flight/IPC.

- Both graphs look similar as the microengines do not do any form of out-of-order speculation. Hence the maximum IPC value is 1 and the maximum number of instructions in flight is 5

Branch Predictions

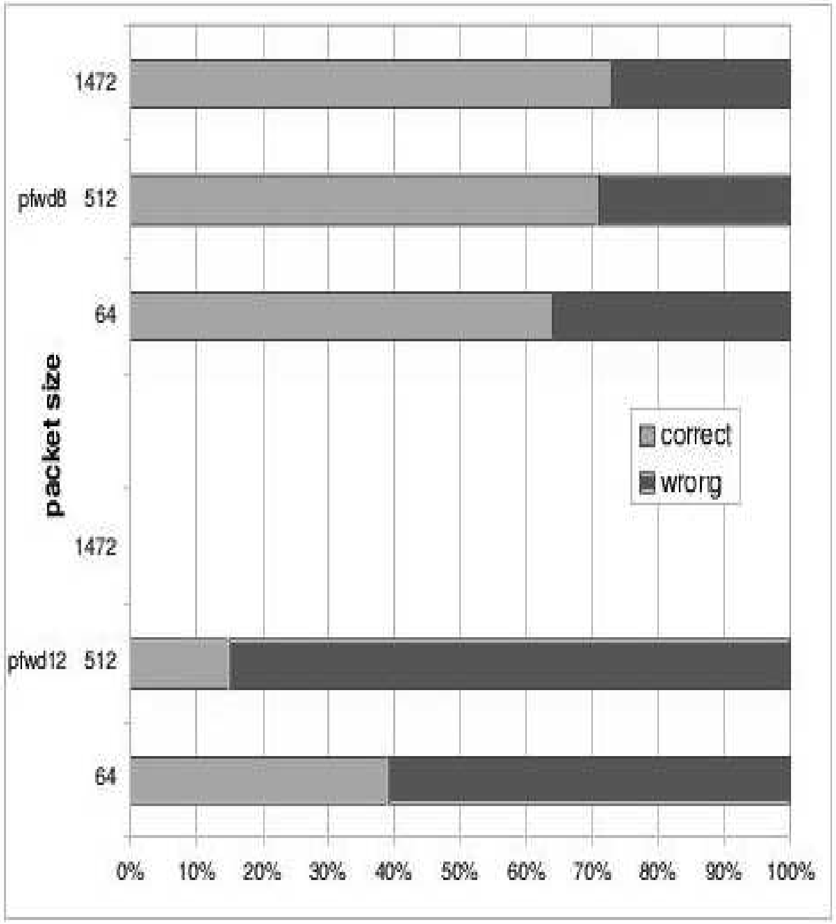
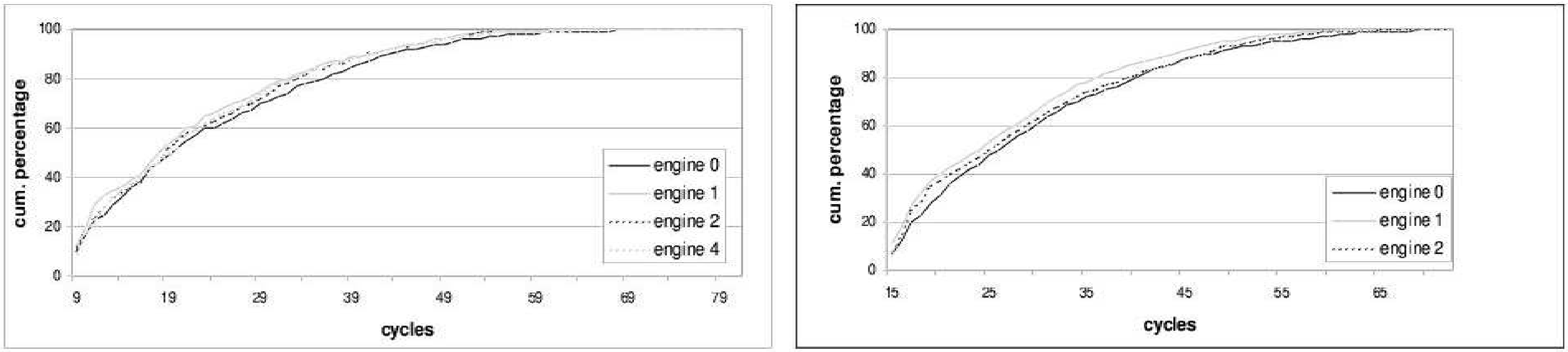


Figure 5: Branch predictions.

- Microengines have 3 mechanism to improve branch predictions
  - Deferred Branches
  - Setting Condition code earlier
  - Branch guessing with programmer assistance
- From graphs, different applications can have totally different branch prediction results. It depends a lot on the programmer

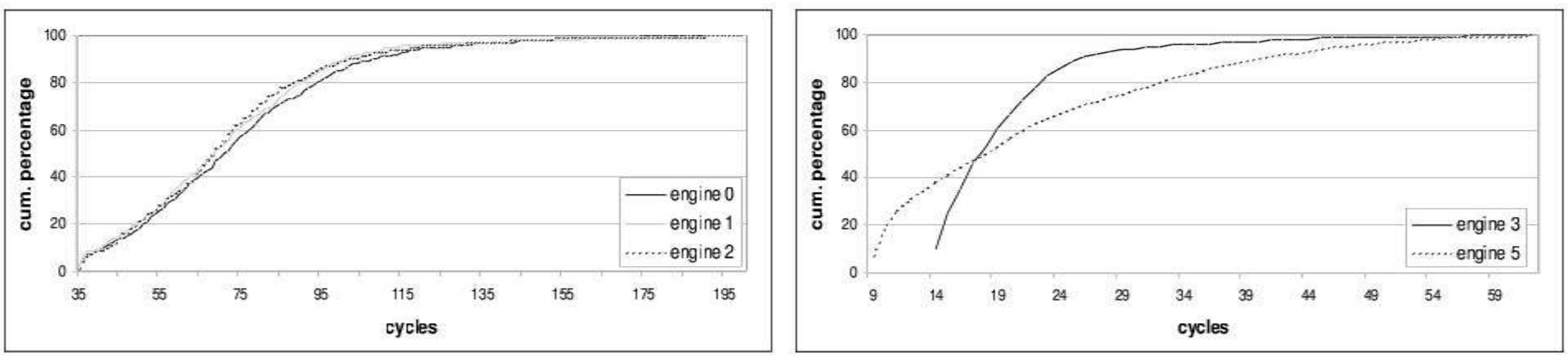
Memory Access Latencies



(a) FBI CSR latency.

(b) Receive FIFO buffer latency.

Figure 6: Latencies for FBI CSR and receive FIFO buffer.

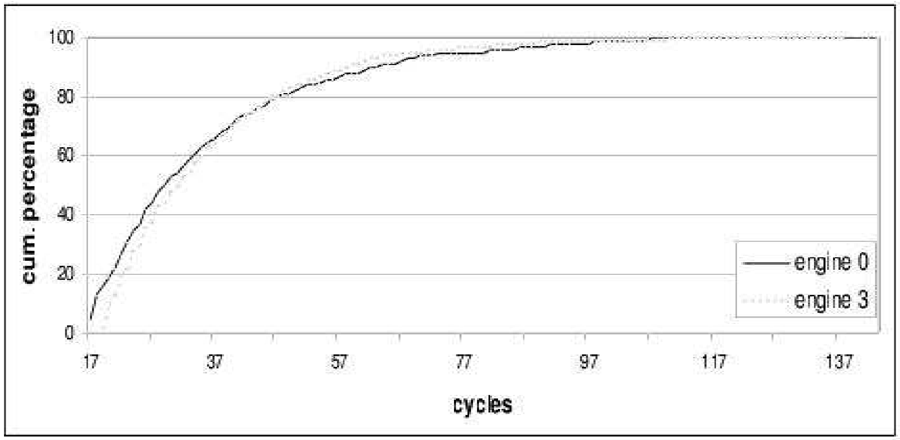


(a) SDRAM latency.

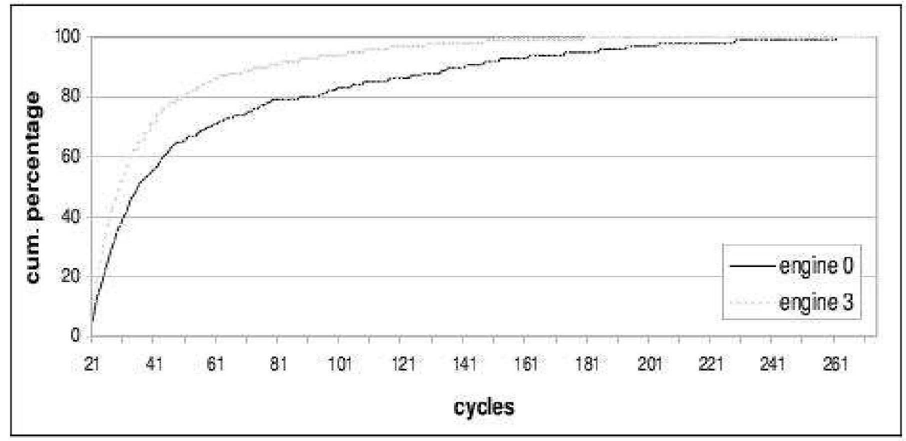
(b) Scratchpad RAM latency.

Figure 7: Latencies for SDRAM and Scratchpad RAM.

Memory Access Latencies (2)



(a) SRAM latency (unlocked).



(b) SRAM latency (locked).

Figure 8: Latencies for SRAM.

- Memory Access Latencies are big (> 50% of accesses to SDRAM take at least 75 cycles for example)
- Latency is hidden by multiple threads on each microengine. When 1 thread is I/O bound, another thread can do useful work on the microengine