Performance Monitoring

(Based on PXA250)

Overview

- Performance Monitoring Registers
  - Clock counter
  - Performance counter
  - Performance monitor control register
- Performance Monitoring Examples
  - CPI (cycles per instruction)
  - I-cache/D-cache efficiency
  - Instruction fetch latency
  - Data/bus request buffer
  - Stall/writeback statistics
  - I-TLB/D-TLB efficiency

Performance Monitoring Registers

- Two 32-bit Performance Monitor Count (PMN0/1)
  - Can monitor two events simultaneously
- One 32-bit Clock Counter (CCNT)
  - Useful when measuring total execution times
- Performance Monitor Control Register (PMNC)
  - Can select monitored events at PMN0 & PMN1
  - Reset, detect overflow of counters
- Accessible through Coprocessor 14 (CP14)
  - PMNC: reg 0
  - CCNT: reg 1 , PMN0/1: regs 2/3

Events Type

- Occurrence events:
  - **`whenever`** the event happens, counter ++
- Duration events:
  - **`while`** (ConditionIsTrue) counter++
- Performance monitoring Examples
  - Clock counter ⇒ total number of cycles during monitoring
  - Performance counter ⇒ Duration events counting
  - % of Event duration =
    (performance counter / clock counter) * 100

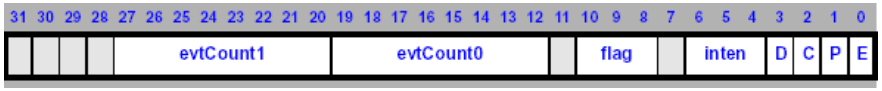
Extending Count Duration > 32 bits

- Overflow interrupts whenever Counter reaches the max value (if enabled)
- Counting continues under Overflow until it is disabled by ISR
- In a typical ISR will do accumulate the number of overflows in a memory location.
- ISR overhead is negligible:
  - Overhead = tens of cycles << 2^{32}

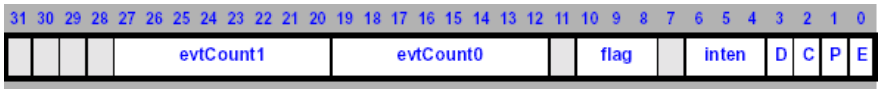
Clock Counter (CCNT)

- Counts core clock cycles.
- If the max value (0xFFFFFFFF)is reached,
  - Reset to 0
  - Overflow bit (PMNC bit 10) set to 1
  - IRQ/FIQ
    - If Enabled (PMNC bit 6)

Performance Monitor Control Reg. (PMNC)

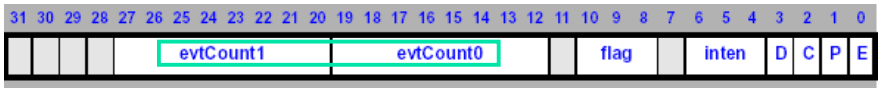


- 19:12/27:20 : Event Type for PMN0/1
- 10:8 : Overflow/Interrupt flag
  - Identifies which counter overflowed
  - bit 10: CCNT, bit 8/9: PMN0/1
- 6:4 : Interrupt Enable
  - Bit 6: CCNT, bit 4/5: PMN0/1
  - 1 = enable , 0 = disable



- 3(D) : clock counter divider
  - 0 = count every cycle , 1 = count every 64 cycles
- 2(C) : clock counter reset
  - 0 = no action , 1 = reset CCNT to 0
- 1(P) : performance counter reset
  - 0 = no action , 1 = reset both PMN0/1 to 0
- 0(E) : enable
  - All three counters disable (0)/enable (1)

Performance Monitoring Event



- 0x0: I-cache miss (O)
- 0x1: I-cache miss or I-TLB miss (D)
- 0x2: stall due to a data dependency (D)
- 0x3: I-TLB miss (O) , 0x4: D-TLB miss (O)
- 0x5: branch executed (O) , 0x6: branch mispredicted (O)
- 0x7: instruction executed (O)
- 0x8/0x9: stall because the data cache buffers are full (D/O)
- 0xA: D-cache access (O) , 0xB: D-cache miss (O)
- 0xC: D-cache write-back (O)
- 0xD: S/W changed the PC (O)

Performance Monitoring Examples

Mode	PMNC.evtCount0	PMNC.evtCount1
Instruction Cache Efficiency	0x7 (instruction count)	0x0 (ICache miss)
Data Cache Efficiency	0xA (Dcache access)	0xB (DCache miss)
Instruction Fetch Latency	0x1 (ICache cannot deliver)	0x0 (ICache miss)
Data/Bus Request Buffer Full	0x8 (DBuffer stall duration)	0x9 (DBuffer stall)
Stall/Writeback Statistics	0x2 (data stall)	0xC (DCache writeback)
Instruction TLB Efficiency	0x7 (instruction count)	0x3 (ITLB miss)
Data TLB Efficiency	0xA (Dcache access)	0x4 (DTLB miss)

Example: I-Cache Efficiency & CPI Monitoring

- PMN0 ⇒ Instruction Count
  - PMNC.evtCount0 = 0x7
- PMN1 ⇒ # of I-cache misses
  - PMNC.evtCount1 = 0x0
- CCNT ⇒ Total number of cycles

- I-cache miss rate = PMN1/PMN0
- CPI (cycles per ins.) = CCNT/PMN0

Configuring PMNC

```
; Configure PMNC with the following values:
;   evtCount0 = 7, evtCount1 = 0instruction cache efficiency
;   inten = 0x7set all counters to trigger an interrupt on
        overflow
;   C = 1   reset CCNT register
;   P = 1   reset PMN0 and PMN1 registers
;   E = 1   enable counting
MOV  R0,#0x7777
MCR  P14,0,R0,C0,C0,0; write R0 to PMNC
; Counting begins
```

Interrupt handling

```
IRQ_INTERRUPT_SERVICE_ROUTINE:
; Assume that performance counting interrupts are the only IRQ in the system
MRC  P14,0,R1,C0,C0,0; read the PMNC register
BIC  R2,R1,#1       ; clear the enable bit
MCR  P14,0,R2,C0,C0,0; clear interrupt flag and disable counting
MRC  P14,0,R3,C1,C0,0; read CCNT register
MRC  P14,0,R4,C2,C0,0; read PMN0 register
MRC  P14,0,R5,C3,C0,0; read PMN1 register

<process the results>
SUBS PC,R14,#4      ; return from interrupt
```

Computing results

```
; Assume CCNT overflowed

CCNT = 0X0000,0020 ;Overflowed and continued counting
Number of instructions executed = PMN0 = 0x6AAA,AAAA
Number of instruction cache miss requests = PMN1 = 0x0555,5555
Instruction Cache miss-rate = 100 * PMN1/PMN0 = 5%
CPI = (CCNT + 2^32)/Number of instructions executed = 2.4 cycles/instruction
```

Example: D-Cache Efficiency

- PMN0 ⇒ # of D-cache accesses
  - PMNC.evtCount0 = 0xA
- PMN1 ⇒ # of D-cache misses
  - PMNC.evtCount1 = 0xB

- D-cache miss rate = PMN1/PMN0