Fundusze Europejskie
Wiedza Edukacja Rozwój

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Społeczny

Politechnika Śląska jako Centrum Nowoczesnego Kształcenia
opartego o badania i innowacje

POWR.03.05.00-IP.08-00-PZ1/17

Projekt współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego

Microprocessor and Embedded Systems

Faculty of Automatic Control, Electronics and Computer Science,
Informatics, Bachelor Degree

Lecture 9

Pentium
Pentium MMX

Bartłomiej Zieliński, PhD, DSc

Pentium (MMX)

Program:

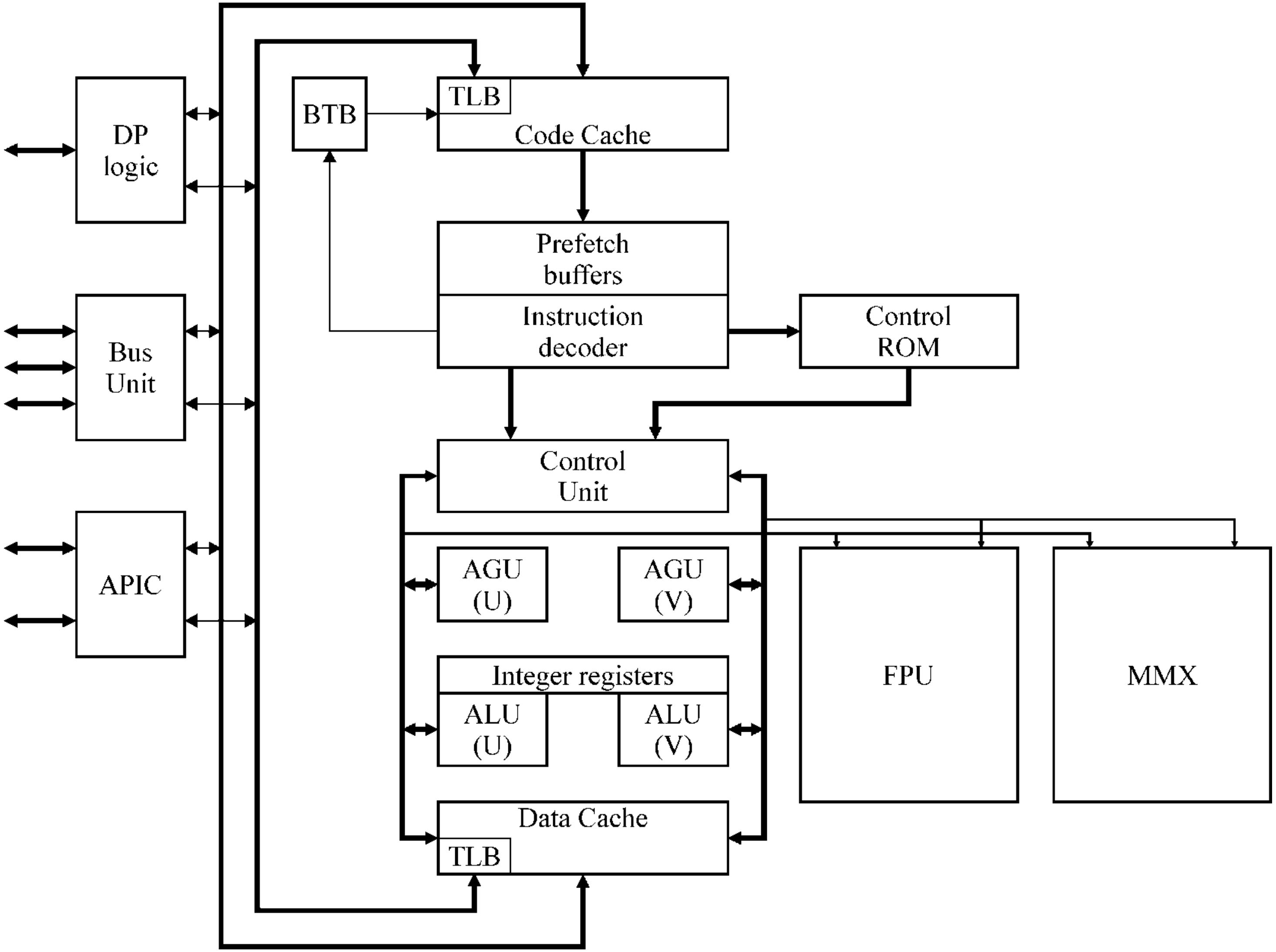
- Pentium processor
  - Structure
  - Operating rules
  - FPU processing
  - APIC & dual processing
- Pentium MMX extension

Pentium (MMX)

- Pentium processor
  - CISC, superscalar
  - Pipes:
    - Integer: U, V – 5 stages, almost synchronous
    - FPU – 8 stages
    - (MMX – 8 stages)
  - MESI cache
    - Separate for code & data
  - BTB: 256 entries
  - No register renaming
  - No Out of order execution

Pentium (MMX)

- Pentium – structure



Pentium (MMX)

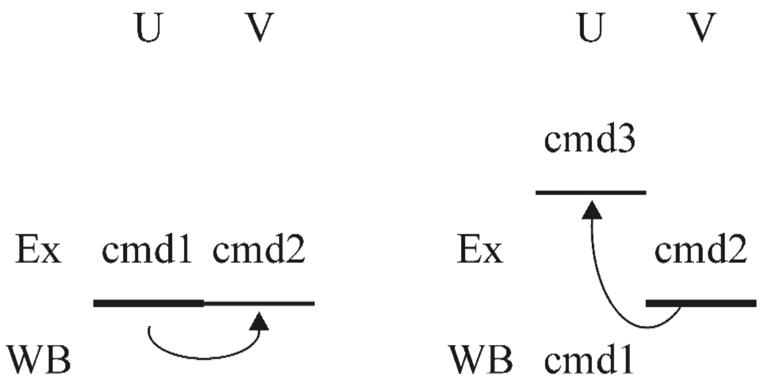
- Pentium – pipelines
  - Pairing rules
    - Both commands must be simple
      - No µcode, 1 clk/instr.
      - E.g., MOV reg, reg; INC; DEC; cond.jmp., ALU (2-3 clk/instr.)
    - No mutual RAW, WAW dependencies
      - (No Register Alias Table)
    - No mutual implicit dependencies
      - Compare-cond.jmp. (dependency on flags)
      - Push/pop (dependency on SP)
      - *But these two are hardware solved*
    - No command contains immediate data & displacement
    - Prefixed commands: only U pipe

Pentium (MMX)

- Pentium – pipeline stages
  - PF (*Prefetch*) – fetch code line from mem or cache
  - F (*Fetch*, MMX only) – instr. len., prefixes decode
  - *Queue to reduce delays in D1 stage if PF/F stop*
  - D1 (*Decode*) – *can two instructions be paired?*
    - Prefix → instr→U, 1 clk lost, no paring
  - D2 (*AddrGen*) – address calculation
    - 1 clk even if displacement & immediate or base+index
  - EX (*Execute*) – mem rd., ALU, branch pred. verify
    - ALU+cache: +1 clk
  - WB (*WriteBack*) – wr. results, branch pred. verify

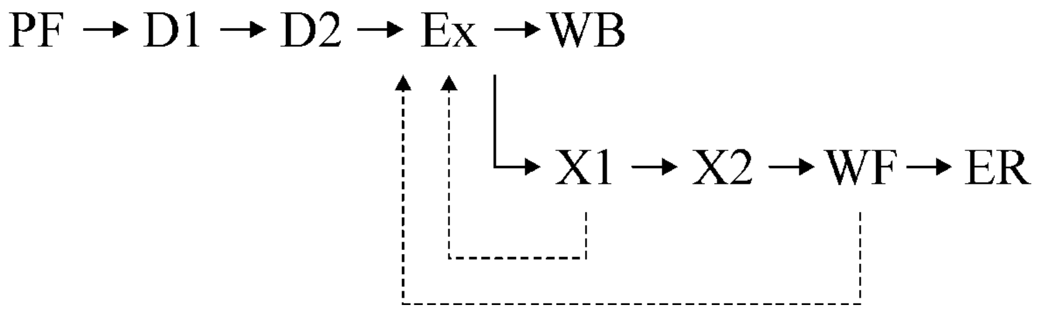
Pentium (MMX)

- Pentium – pipeline synchronism
  - D1, D2 – fully synchronous
    - U stop → V stop, V stop → U stop
  - EX – not fully synchronous
    - Pipe stops if cache miss
    - U stop → V stop
    - V stop → U go,
      - But both must reach WB to allow next instr. enter EX



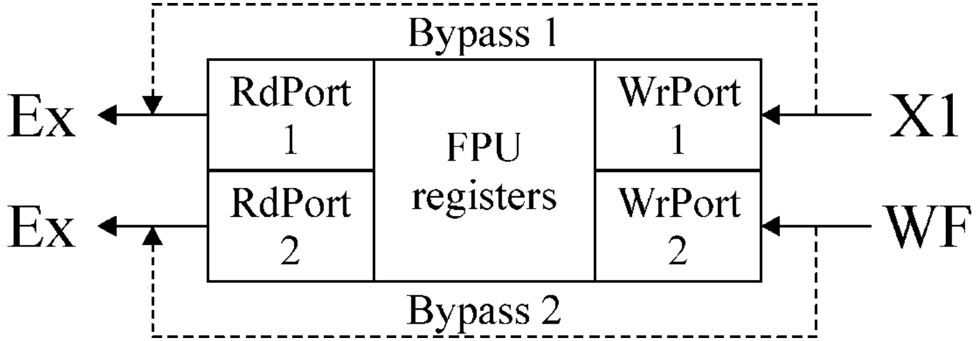
Pentium (MMX)

- Pentium – FPU pipeline
  - Partially common
    - PF, (F,) D1, D2, Ex
      - Like U pipe
    - X1 – FPU #1
      - Mem data conversion to FPU format
      - Write operand to FPU regs
      - Bypass 1
    - X2 – FPU #2
    - WF – Write FPU
      - Bound results, write to FPU regs.
      - Bypass 2
    - ER – error reporting
      - Status word update



Pentium (MMX)

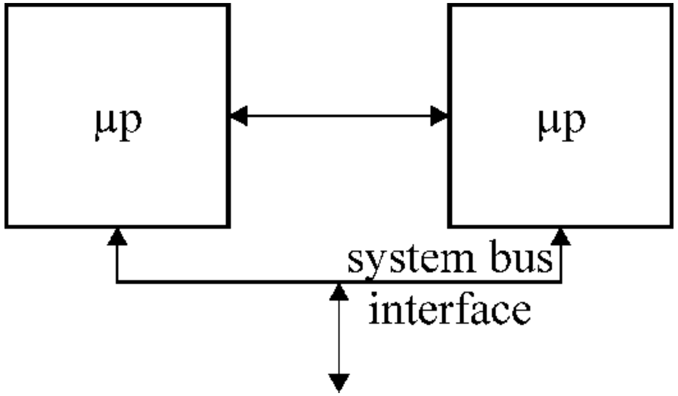
- Pentium – FPU pipeline
  - Bypass 1, 2



  - Safe Instruction Recognition
    - Early (X1) recognition if overflow, exception etc. is likely to occur
      - If safe, next command may complete EX
      - Otherwise, next command waits until „unsafe” completes ER
        - » Even if nothing happened
        - » 4 clk delay

Pentium (MMX)

- Pentium – multiprocessing



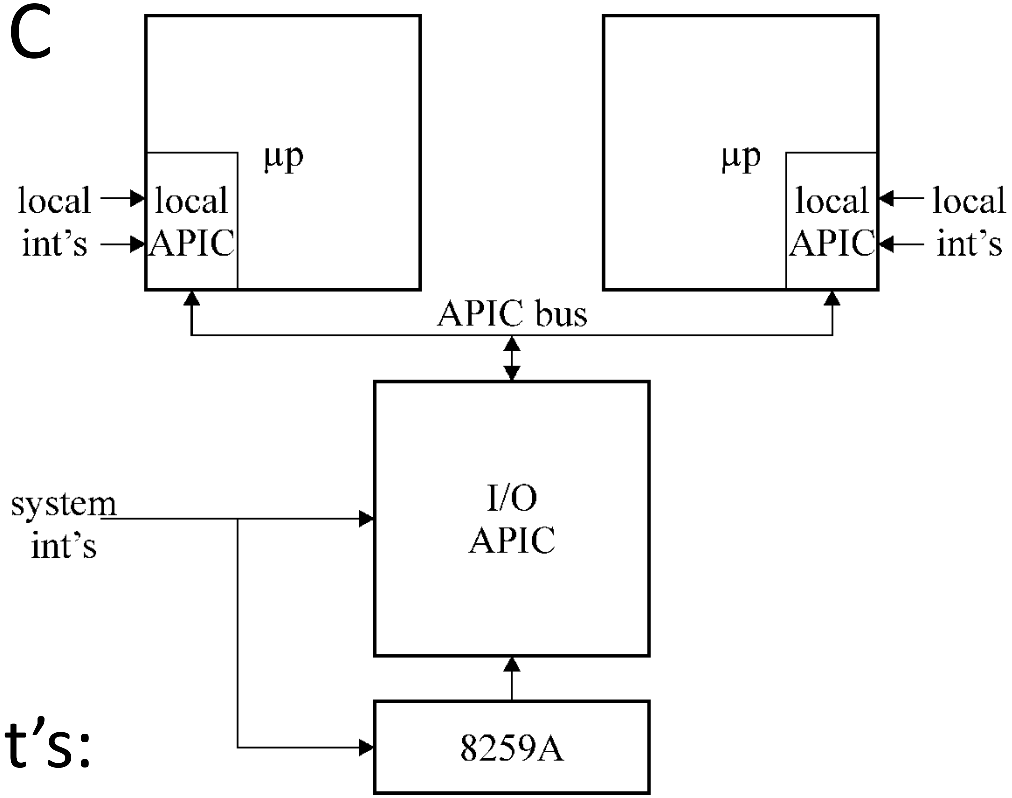
  - Like 1 μp for system
  - Fair bus switching
    - (Like DMA – single transfer mode)
  - Cache – MESI protocol
  - APIC must be turned on

Pentium (MMX)

- Pentium – APIC
  - PIC with multiprocessing support
    - In multi-μp can be used with I/O APIC
    - Can work as the only PIC
    - Can be blocked
    - Can be 8259A-compatible
    - Contains 32-b timer for local timer interrupts

Pentium (MMX)

- Pentium – APIC



  - µp accepts int's:
    - Locally via LINT0, LINT1
    - System via APIC bus
    - From other µp's via APIC bus

Pentium (MMX)

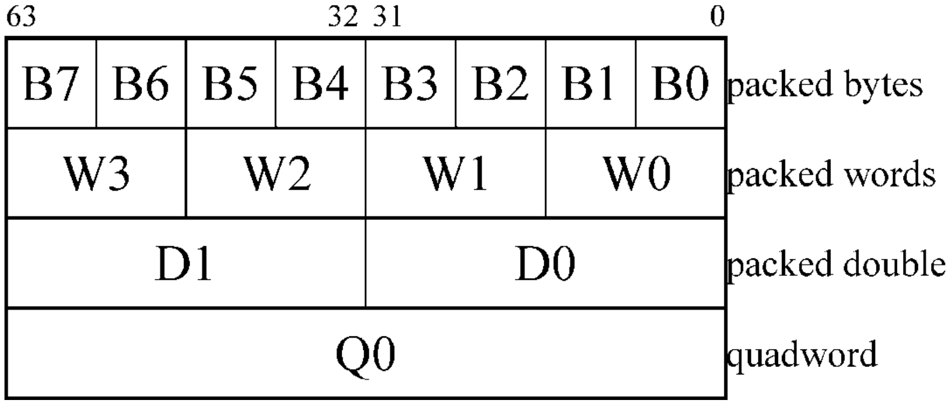
- Pentium – APIC
  - I/O APIC
    - Capture system int’s
    - Distributes system int’s to proper µp’s
    - Programmable schemes of int distribution to µp’s
  - Operating modes:
    - *Normal*: local APIC soft- & hardware enabled
    - *Bypass*: as if no APIC present; LINT → INT, NMI
      - No multiprocessing
    - *Through local*: multiµp; extern. int’s through local APIC
      - INT from APIC bus is possible
    - *Masked*: software disabled; APIC bus int’s masked
      - local APIC ignores LINT

Pentium (MMX)

- Pentium – MMX extension
  - *MultiMedia eXtensions*
    - For multimedia (and not only!) processing
    - SIMD (*Single Instruction Multiple Data*) technique
      - Multiple independent data streams processed exactly the same way by the same commands, at the same time
    - new registers necessary
      - $MM_{0..7}$ – hidden „under" FPU registers
        - » Old OS did not know about their presence
          - problems in multitasking
        - » Long switch between FPU and MMX
        - » MMX/FPU operations should be grouped

Pentium (MMX)

- Pentium – MMX extension
  - Registers, data types



63 32 31 0

B7 B6 B5 B4 B3 B2 B1 B0 packed bytes

W3 W2 W1 W0 packed words

D1 D0 packed double

Q0 quadword

  - Operations
    - Data transfer
    - Packing/unpacking
    - Arithmetical (+, -, ×, ×+)
    - Logical (and, nand, or, xor)
    - Shift, compare
    - Exit MMX

Pentium (MMX)

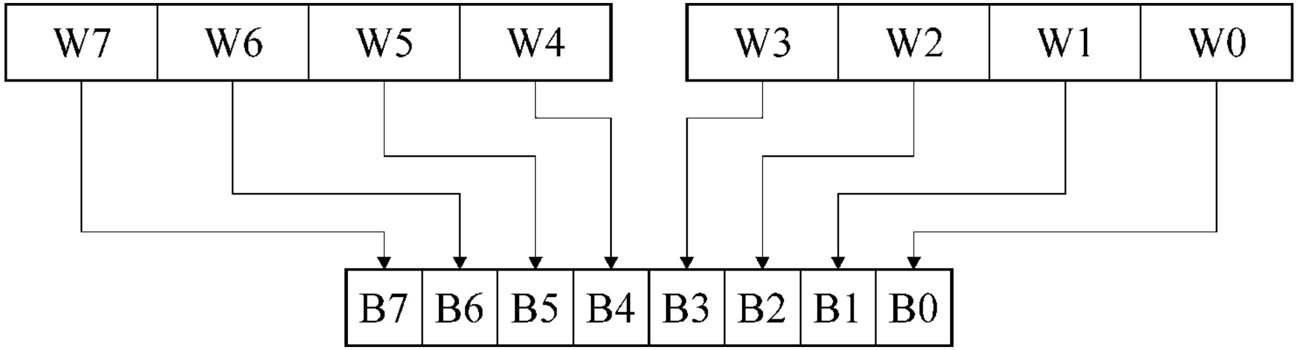
- Pentium – MMX extension
  - 57 new commands
    - Mnemonic code:
      - **p** – if packed data type
      - Operation code, e.g., **add**, **sub**, **cmp**, etc.
      - Suffix:
        - » **us** (*unsigned saturation*)
        - » **s** (*signed saturation*)
        - » **b** (*packed byte*)
        - » **w** (*packed word*)
        - » **d** (*packed double word*)
        - » **q** (*quadword*)

Pentium (MMX)

- Pentium – MMX extension
  - 57 new commands
    - Saturation
      - Carry ignore
        - Cut result to possible values range
          - If result < min
            - Result = min
          - If result > max
            - Result = max
          - Otherwise no result cut
      - Usable in multimedia processing, e.g.:
        - Graphics processing
        - Sound processing

Pentium (MMX)

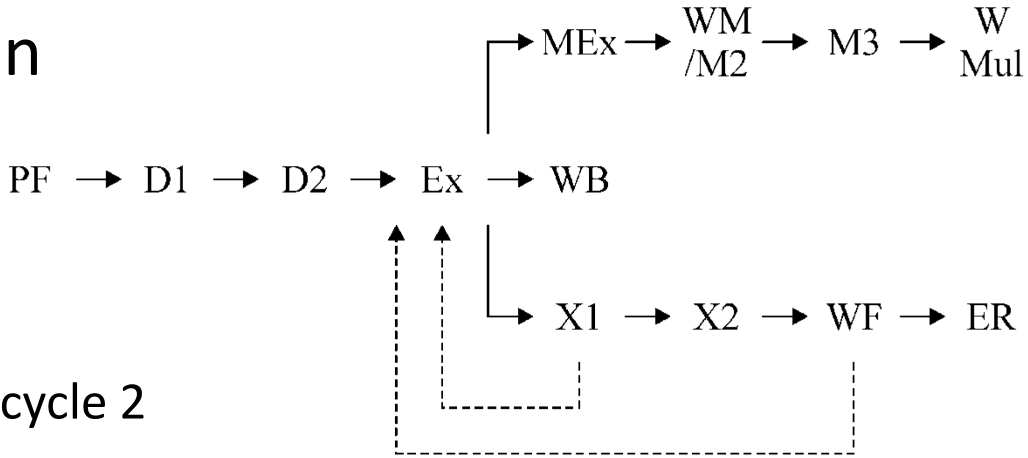
- Pentium – MMX extension
  - Example: packswb
    - **P**ack, with **s**aturation, **w**ords→**b**ytes



```
for i=0 to 7
   if W[i] > 127
      B[i] = 127;
   else if W[i] < -128
      B[i] = -128;
   else
      B[i] = Lo (W[i]);
   end if
end for
```

Pentium (MMX)

- Pentium – MMX extension
  - Pipelined execution
    - MEx
      - MMX exec #1
    - WM/M2
      - Write MMX/Mul cycle 2
    - M3
      - Mul cycle 3
    - WMul
      - Write mul result
    - Each phase – 1 clk



Pentium (MMX)

- Pentium – MMX extension
  - Pairing
    - MMX/MMX if different MM registers used
    - MMX/ALU
    - But not MMX/FPU