CPU Structure and Function

Chapter 12
Processor Organization

- CPU must:
  - Fetch instructions
  - Interpret instructions
  - Fetch data
  - Process data
  - Write data
CPU With Systems Bus

ALU

Registers

Control Unit

Control Bus  Data Bus  Address Bus

System Bus

Introduction to Computer Architecture and Organization

Lesson 17 – Slide 3/36
CPU Internal Structure

Arithmetic and Logic Unit
- Status Flags
- Shifter
- Complementer

Arithmetic and Boolean Logic

Internal CPU Bus

Registers

Control Unit

Control Paths
Registers

- CPU must have some working space (temporary storage)
- Called registers
- Number and function vary between processor designs
- One of the major design decisions
- Top level of memory hierarchy
- Registers:
  - User-visible registers
  - Control and status registers
User-Visible Registers

• General Purpose
• Data
• Address
• Condition Codes
General Purpose Registers (1)

- May be true general purpose
- May be restricted
- May be used for data or addressing
  - Data
    - Accumulator
  - Addressing
    - Segment
General Purpose Registers (2)

• Make them general purpose
  – Increase flexibility and programmer options
  – Increase instruction size & complexity

• Make them specialized
  – Smaller (faster) instructions
  – Less flexibility
How Many GP Registers?

- Between 8 - 32
- Fewer = more memory references
- More does not reduce memory references and takes up processor real estate
- But in RISC processors, hundreds of registers
How big?

- Large enough to hold full address
- Large enough to hold full word
- Often possible to combine two data registers
  - C programming
  - double int a;
  - long int a;
Condition Code Registers

- **Flags**
- Sets of individual bits
  - e.g. result of last operation was zero
- Can be read (implicitly) by programs
  - e.g. Jump if zero
- Can not (usually) be set by programs
Control & Status Registers

• For instruction execution:
  – Program Counter (PC)
  – Instruction Register (IR)
  – Memory Address Register (MAR)
  – Memory Buffer Register (MBR)
Program Status Word (PSW)

- A set of bits
- Includes Condition Codes
- Sign of last result
- Zero
- Carry
- Equal
- Overflow
- Interrupt enable/disable
- Supervisor
Supervisor Mode

- Kernel mode
- Allows privileged instructions to execute
- Used by operating system
- Not available to user programs
Other Registers

• May have registers pointing to:
  – Process control blocks
  – Interrupt Vectors

• CPU design and operating system design are closely linked
### Example Register Organizations

#### Data Registers
- D0
- D1
- D2
- D3
- D4
- D5
- D6
- D7

#### General Registers
- AX: Accumulator
- BX: Base
- CX: Count
- DX: Data
- EAX
- EBX
- ECX
- EDX

#### Pointer & Index
- ESP
- SP: Stack Pointer
- EBP
- BP: Base Pointer
- ESI
- SI: Source Index
- EDI
- DI: Dest Index

#### Address Registers
- A0
- A1
- A2
- A3
- A4
- A5
- A6
- A7
- A7'

#### Program Status
- FLAGS Register
- Instruction Pointer

(c) 80386 - Pentium 4

(b) 8086

(a) MC68000

---

**Introduction to Computer Architecture and Organization**

*Lesson 17 – Slide 16/36*
Instruction Cycle

• Subcycles:
  – **Fetch**: Read the next instruction from memory into the CPU
  – **Execute**: Interpret the opcode and perform the indicated operation
  – **Interrupt**: If interrupts are enabled and an interrupt has occurred, save the current process state and service the interrupt
Indirect Cycle

- May require memory access to fetch operands
- Indirect addressing requires more memory accesses
- Can be thought of as additional instruction subcycle
Instruction Cycle State Diagram

Introduction to Computer Architecture and Organization
Lesson 17 – Slide 20/36
Data Flow (Instruction Fetch)

- Depends on CPU design
- In general:

- Fetch
  - PC contains address of next instruction
  - Address moved to MAR
  - Address placed on address bus
  - Control unit requests memory read
  - Result placed on data bus, copied to MBR, then to IR
  - Meanwhile PC incremented by 1
Data Flow (Data Fetch)

- IR is examined
- If indirect addressing, indirect cycle is performed
  - Right most N bits of MBR transferred to MAR
  - Control unit requests memory read
  - Result (address of operand) moved to MBR
Data Flow (Fetch Diagram)

MBR = Memory buffer register
MAR = Memory address register
IR = Instruction register
PC = Program counter
Data Flow (Indirect Diagram)
Data Flow (Execute)

- May take many forms
- Depends on instruction being executed
- May include
  - Memory read/write
  - Input/Output
  - Register transfers
  - ALU operations
Data Flow (Interrupt)

- Simple
- Predictable
- Current PC saved to allow resumption after interrupt
- Contents of PC copied to MBR
- Special memory location (e.g. stack pointer) loaded to MAR
- MBR written to memory
- PC loaded with address of interrupt handling routine
- Next instruction (first of interrupt handler) can be fetched
Data Flow (Interrupt Diagram)
Prefetch

- Fetch accessing main memory
- Execution usually does not access main memory
- Can fetch next instruction during execution of current instruction
- Called instruction prefetch
Improved Performance

• But not doubled:
  – Fetch usually shorter than execution
  – Any jump or branch means that prefetched instructions are not the required instructions

• Add more stages to improve performance
Pipelining

- Fetch instruction (FI)
- Decode instruction (DI)
- Calculate operands (CO)
- Fetch operands (FO)
- Execute instructions (EI)
- Write operand (WO)

- Overlap these operations
Two Stage Instruction Pipeline

(a) Simplified view

Instruction → Fetch → Instruction → Execute → Result

Instruction → Fetch → Instruction → Execute → Result → Discard

New address

Wait

Wait
Timing of Pipeline

Introduction to Computer Architecture and Organization
Lesson 17 – Slide 32/36
Branch in a Pipeline

Introduction to Computer Architecture and Organization

Lesson 17 – Slide 33/36
Six Stage Instruction Pipeline
Alternative Pipeline Depiction

(a) No branches

(b) With conditional branch
Speedup Factors with Instruction Pipelining

(a) Speedup factor vs. Number of instructions for different stages:
- k = 12 stages
- k = 9 stages
- k = 6 stages

(b) Speedup factor vs. Number of stages for different instructions:
- n = 30 instructions
- n = 20 instructions
- n = 10 instructions