Chapter 3
System Buses
**Program Concept**

- Hardwired systems are inflexible.
- General purpose hardware can do different tasks, given correct control signals.
- Instead of re-wiring, supply a new set of control signals.
What is a program?

• A sequence of steps
• For each step, an arithmetic or logical operation is done
• For each operation, a different set of control signals is needed
Function of Control Unit

• For each operation a unique code is provided
  —e.g. ADD, MOVE

• A hardware segment accepts the code and issues the control signals

• We have a computer!
Components

• The Control Unit and the Arithmetic and Logic Unit constitute the Central Processing Unit

• Data and instructions need to get into the system and results out
  — Input/output

• Temporary storage of code and results is needed
  — Main memory
Computer Components: Top Level View

CPU
- PC
- IR
- MBR
- I/O AR
- I/O BR

System Bus

Main Memory
- Instruction
- Instruction
- Instruction
- Data
- Data
- Data

Buffers

Legend:
- PC = Program counter
- IR = Instruction register
- MAR = Memory address register
- MBR = Memory buffer register
- I/O AR = Input/output address register
- I/O BR = Input/output buffer register
Instruction Cycle

- Two steps:
  - Fetch
  - Execute
Fetch Cycle

- Program Counter (PC) holds address of next instruction to fetch
- Processor fetches instruction from memory location pointed to by PC
- Increment PC
  - Unless told otherwise
- Instruction loaded into Instruction Register (IR)
- Processor interprets instruction and performs required actions
Execute Cycle

- Processor-memory
  - Data transfer between CPU and main memory

- Processor I/O
  - Data transfer between CPU and I/O module

- Data processing
  - Some arithmetic or logical operation on data

- Control
  - Alteration of sequence of operations
    - e.g. jump

- Combination of above
Example of Program Execution
Instruction Cycle - State Diagram

- Instruction fetch
  - Instruction address calculation
    - Instruction complete, fetch next instruction
  - Instruction operation decoding
- Operand fetch
  - Multiple operands
  -_operand address calculation
    - Return for string or vector data
- Data Operation
  - Operand store
    - Multiple results
Interrupts

- Mechanism by which other modules (e.g. I/O) may interrupt normal sequence of processing
- Program
  - e.g. overflow, division by zero
- Timer
  - Generated by internal processor timer
  - Used in pre-emptive multi-tasking
- I/O
  - from I/O controller
- Hardware failure
  - e.g. memory parity error
Program Flow Control

(a) No interrupts
(b) Interrupts; short I/O wait
(c) Interrupts; long I/O wait
Interrupt Cycle

- Added to instruction cycle
- Processor checks for interrupt
  - Indicated by an interrupt signal
- If no interrupt, fetch next instruction
- If interrupt pending:
  - Suspend execution of current program
  - Save context
  - Set PC to start address of interrupt handler routine
  - Process interrupt
  - Restore context and continue interrupted program
Transfer of Control via Interrupts
Instruction Cycle with Interrupts

START → Fetch Next Instruction → Execute Instruction → Check for Interrupt; Process Interrupt → HALT

Fetch Cycle   Execute Cycle   Interrupt Cycle

Interrupts Disabled

Interrupts Enabled
Program Timing
Short I/O Wait

(a) Without interrupts

(b) With interrupts
Program Timing
Long I/O Wait

(a) Without interrupts

(b) With interrupts
Instruction Cycle (with Interrupts) - State Diagram

Instruction fetch → Instruction address calculation → Instruction operation decoding → Operand fetch → Multiple operands → Operand address calculation → Data Operation → Operand address calculation → Operand store → Multiple results → Interrupt check → Interrupt

Instruction complete, fetch next instruction → Return for string or vector data → No interrupt
Multiple Interrupts

• Disable interrupts
  — Processor will ignore further interrupts whilst processing one interrupt
  — Interrupts remain pending and are checked after first interrupt has been processed
  — Interrupts handled in sequence as they occur

• Define priorities
  — Low priority interrupts can be interrupted by higher priority interrupts
  — When higher priority interrupt has been processed, processor returns to previous interrupt
Multiple Interrupts - Sequential
Multiple Interrupts – Nested
Time Sequence of Multiple Interrupts
Connecting

- All the units must be connected
- Different type of connection for different type of unit
  - Memory
  - Input/Output
  - CPU
Memory Connection

- Receives and sends data
- Receives addresses (of locations)
- Receives control signals
  - Read
  - Write
  - Timing
Input/Output Connection(1)

• Similar to memory from computer’s viewpoint

• Output
  — Receive data from computer
  — Send data to peripheral

• Input
  — Receive data from peripheral
  — Send data to computer
Input/Output Connection(2)

• Receive control signals from computer
• Send control signals to peripherals
  — e.g. spin disk
• Receive addresses from computer
  — e.g. port number to identify peripheral
• Send interrupt signals (control)
CPU Connection

- Reads instruction and data
- Writes out data (after processing)
- Sends control signals to other units
- Receives (& acts on) interrupts