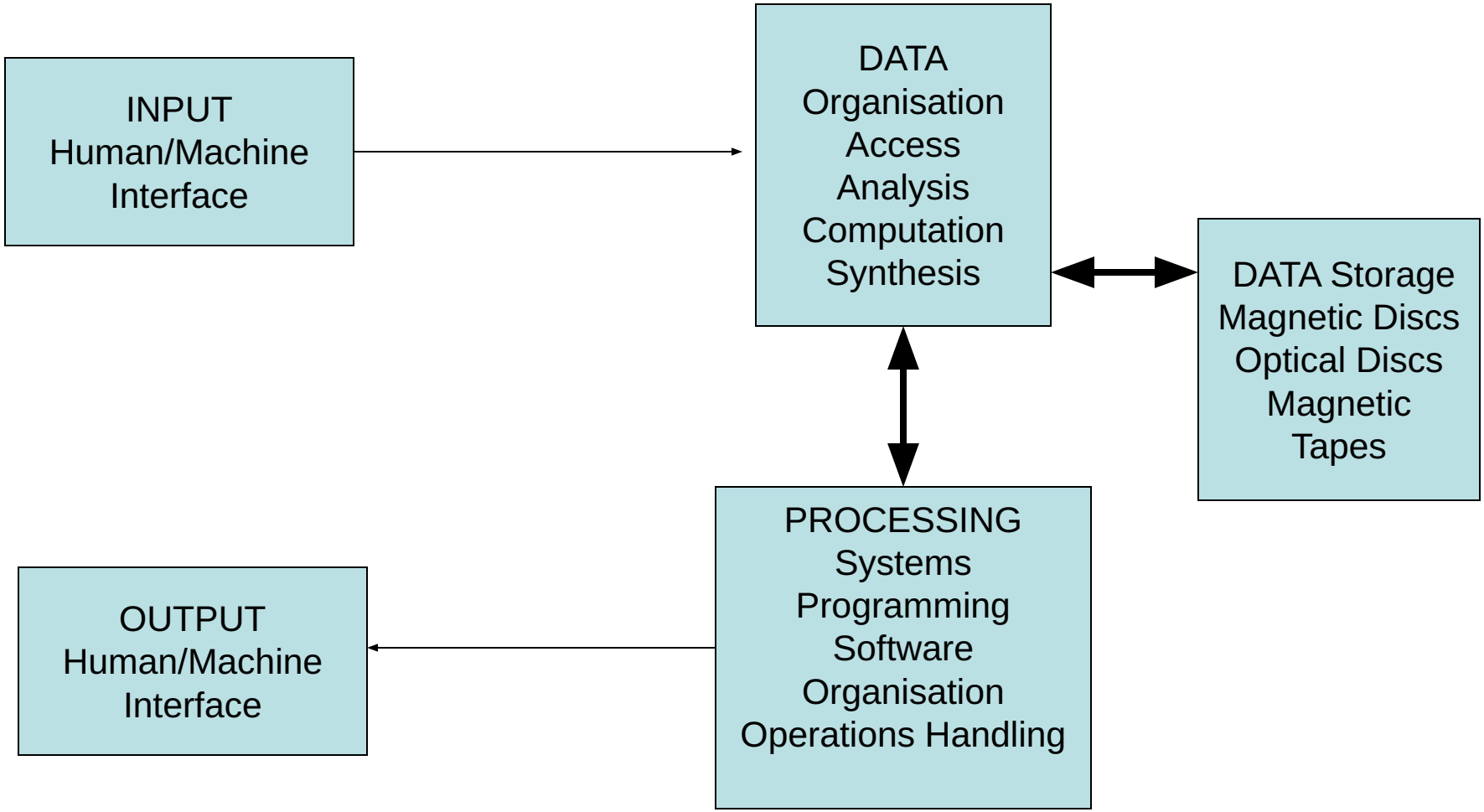
Introduction to Computer Systems

CCE 1011


INPUT
Human/Machine
Interface
DATA
Organisation
Access
Analysis
Computation
Synthesis
DATA Storage
Magnetic Discs
Optical Discs
Magnetic
Tapes
PROCESSING
Systems
Programming
Software
Organisation
Operations Handling
OUTPUT
Human/Machine
Interface

Human Machine Interface

Machine is electronic
Machine is binary

Human input/output
Five senses:
vision
Hearing and speech
Touch
Taste
Smell

Special input/output units to adapt and convert between humans
and machine

Electronic Digital System

Requires electronic building blocks for operation.
Requires boolean mathematics for analysis and design.
Requires systems methods of interpreting binary sequences as

either data
or program instruction

Data Processing

The computer is a state sequential machine.
States give rise to machine cycles.
Machine cycles make up machine instructions.
Systems programming interface higher level program languages to the machine instructions.
Millions of instructions per second can be handled.
Processor architecture handles simultaneous instruction processing using pipelining or parallel paths.
Systems organisation handles movement of data and instructions
in and out of the processing unit
Data codes and standards interpret binary sequences into characters, pixels, audio, etc depending on the I/O device

Data Storage

Volatile storage in RAM – order of few GB
Permanent Read/Write Storage – Hard Disk hundreds of GB
Transportable Storage – CD hundreds of MB
-- DVD tens of GB
-- USB storage device GB
Database Management Systems (DBMS) to store and access data quickly and reliably.
Data storage includes text, vision, audio, general multimedia.

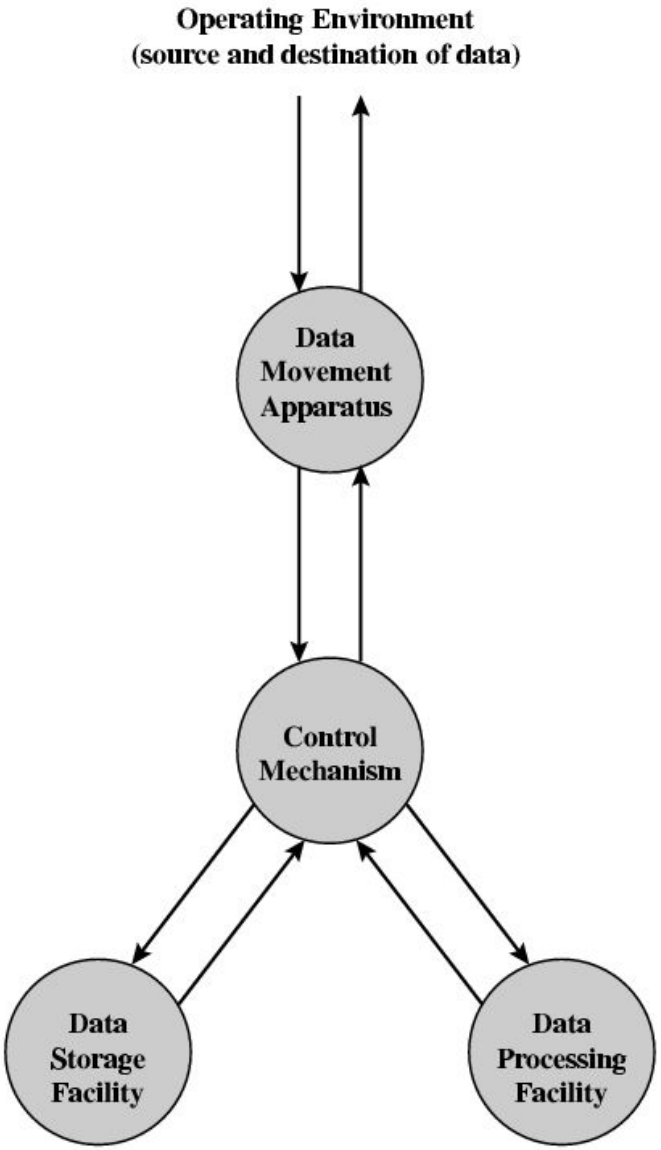
Data Communications

Processing possible on remote machines.
Client-Server Model – Local Area Networks (LAN)
Virtual worldwide processing via Internet.
Personal Multimedia access via wireless electronic equipment – mobiles, wireless LAN's, GSM.
Voice, pictures, videos and data streaming in real-time using the same common backbone telecommunications infrastructure.

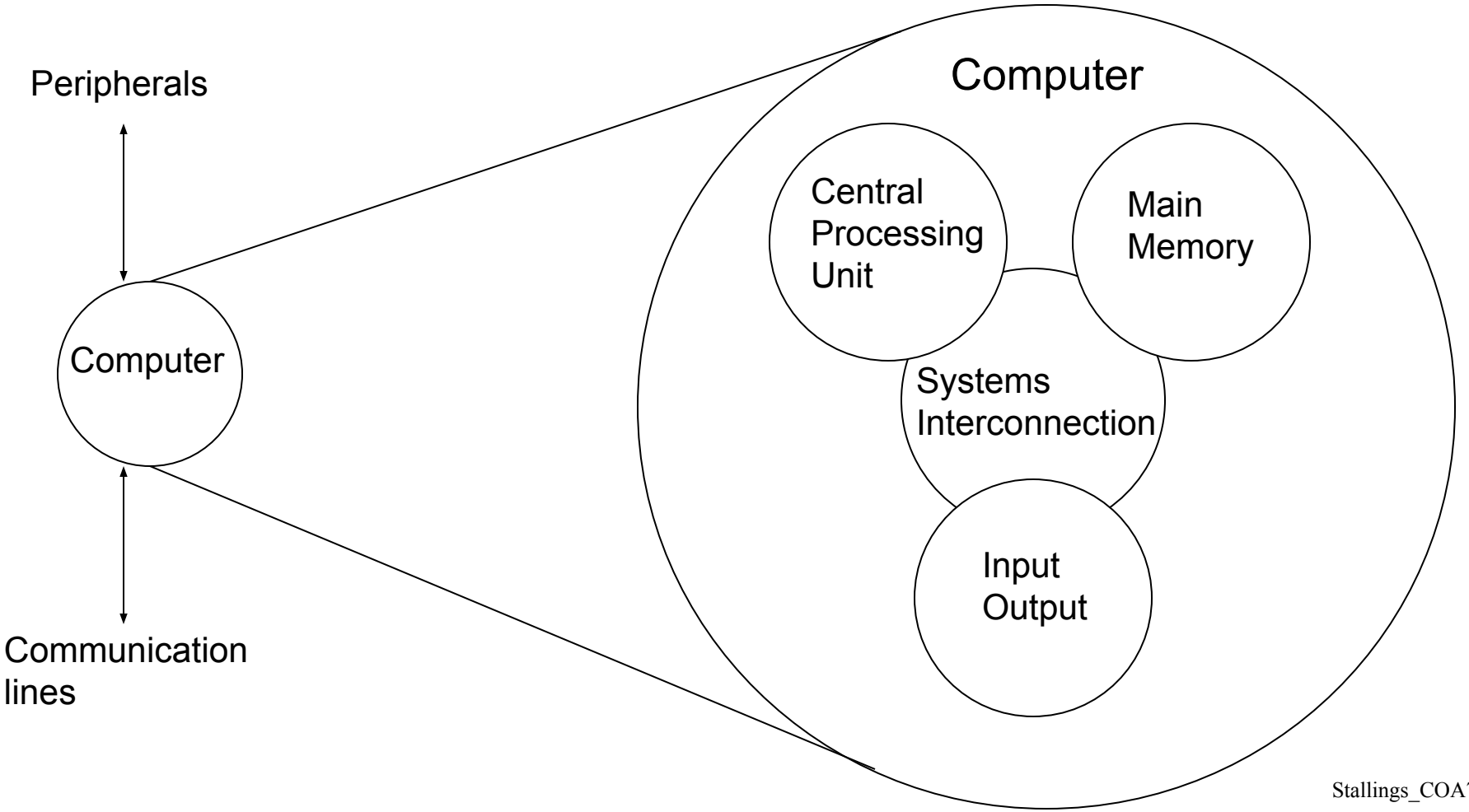
Architecture & Organization 1

- Architecture is those attributes visible to the programmer
  - Instruction set, number of bits used for data representation, I/O mechanisms, addressing techniques.
  - e.g. Is there a multiply instruction?
- Organization is how features are implemented
  - Control signals, interfaces, memory technology.
  - e.g. Is there a hardware multiply unit or is it done by repeated addition?

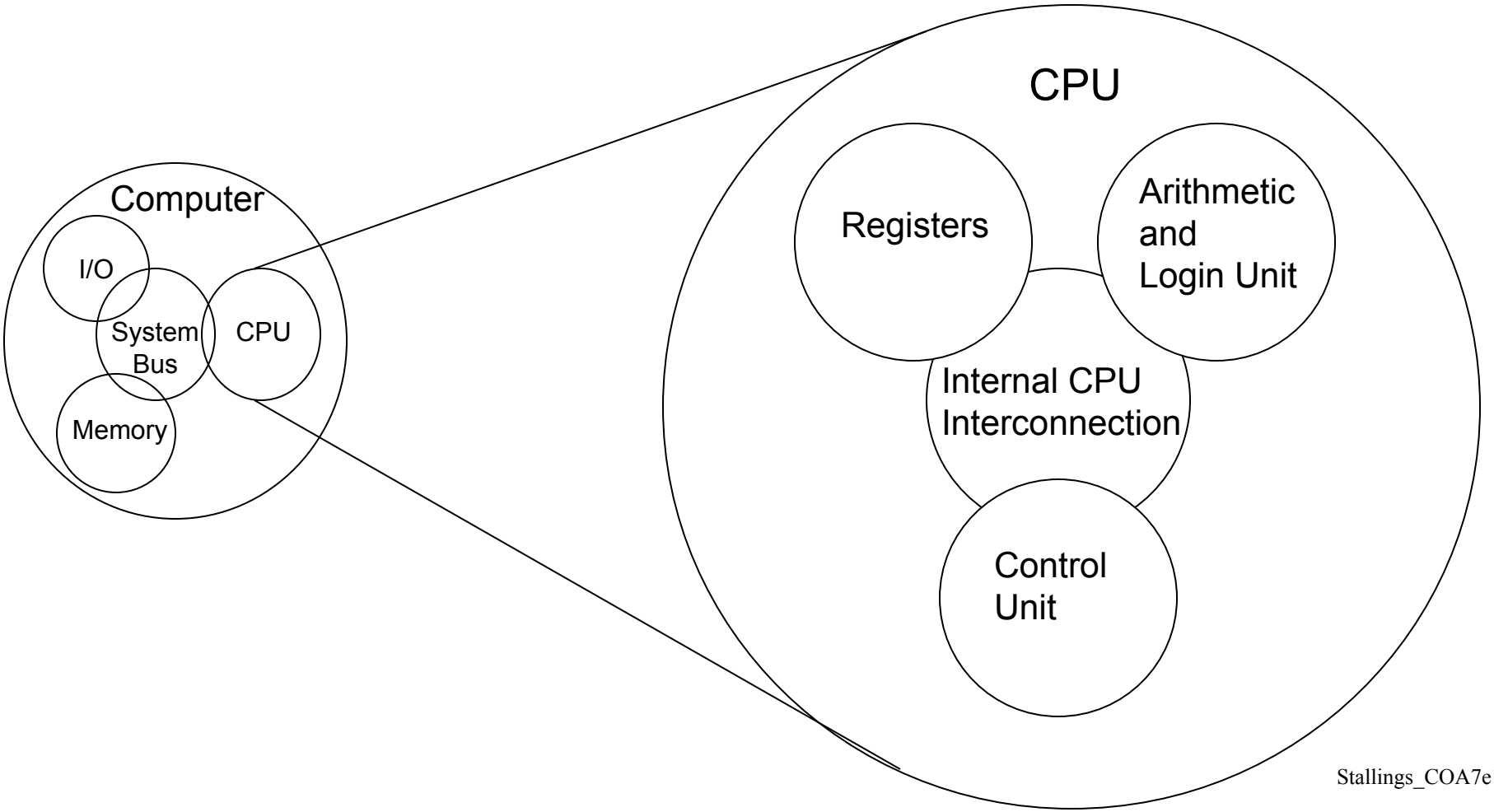
Functional View



Structure - Top Level



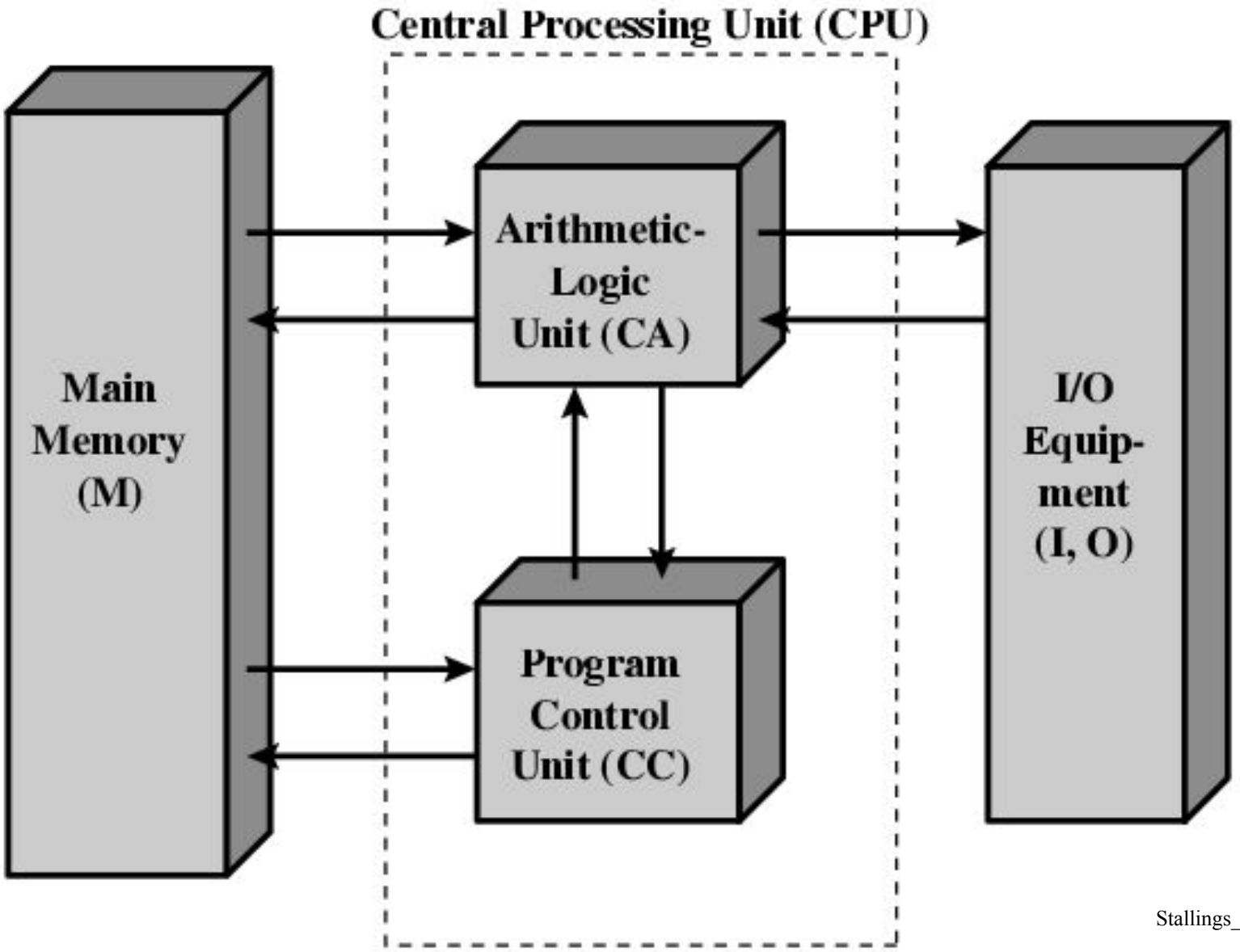
Structure - The CPU



von Neumann/Turing

- Stored Program concept
- Main memory storing programs and data
- ALU operating on binary data
- Control unit interpreting instructions from memory and executing
- Input and output equipment operated by control unit
- Princeton Institute for Advanced Studies
  - IAS
- Completed 1952

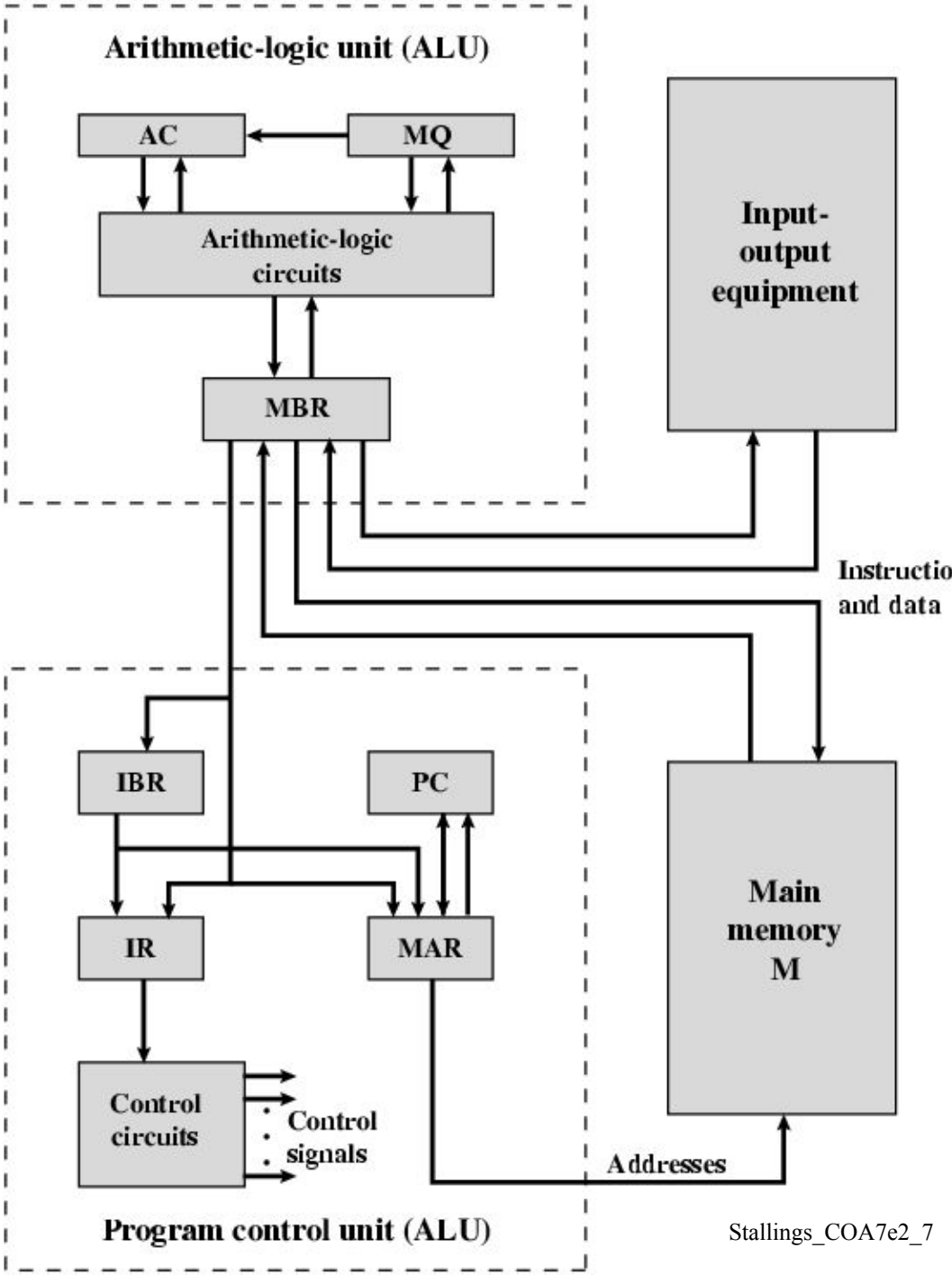
Structure of von Neumann machine



IAS - details

- 1000 x 40 bit words
  - Binary number
  - 2 x 20 bit instructions
- Set of registers (storage in CPU)
  - Memory Buffer Register
  - Memory Address Register
  - Instruction Register
  - Instruction Buffer Register
  - Program Counter
  - Accumulator
  - Multiplier Quotient

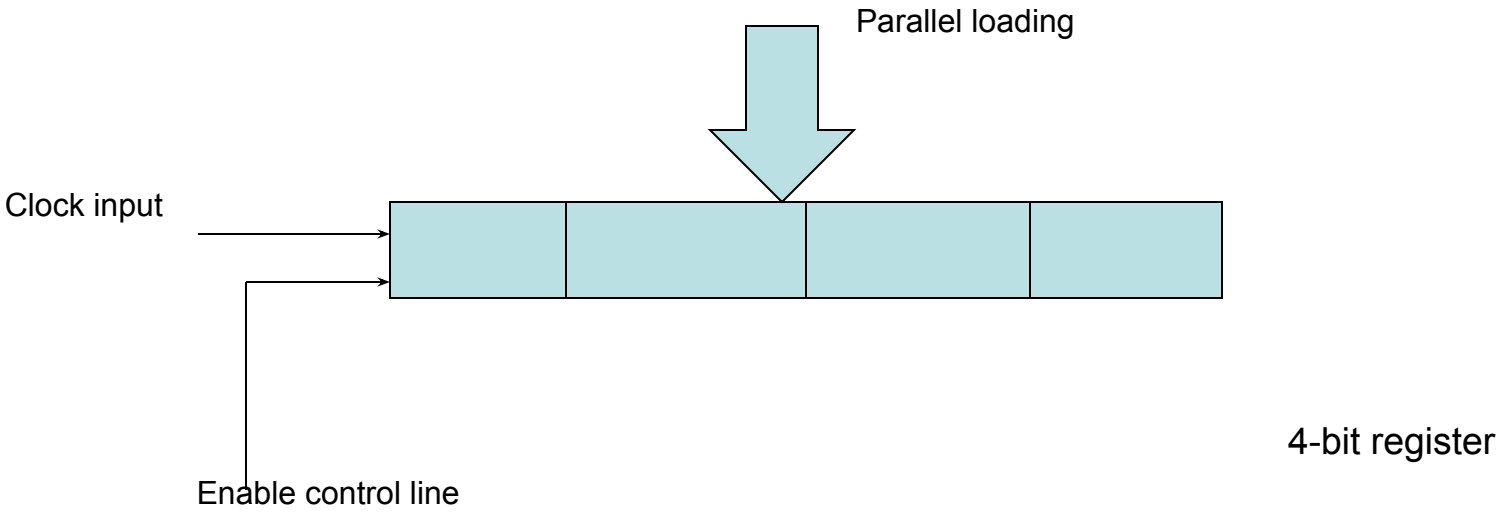
Structure of IAS – detail



Special Registers

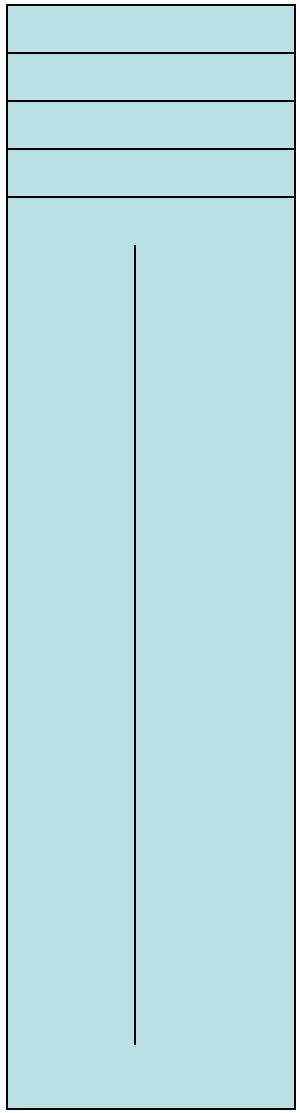
- MAR – Memory address Register is a processor register that contains the address of the memory location being or to be accessed
- MBR – Memory Buffer (Data) Register is a processor register that contains the data to be written to memory or where the data read from memory is stored
- PC – Program Counter is a processor register that contains the start address in memory of the next instruction to be executed
- IR – Instruction Register contains a copy of the instruction operand fetched from memory
- ACC – Accumulator is a register associated with the ALU that is used as a working register to store operands that are to be used in the ALU.

A register is a group of binary storage cells or flipflops. Registers usually have parallel load enable and are synchronised in their timing operation by a clock input



The 4-bit register above can have its four cells loaded in parallel, if the Enable is active and when the clock is active.
Computer registers started as 4-bit registers and today they are 64-bit Registers.

Memory



A memory location is distinguished by an address. At every location there are a number of bits stored as one word. Memory is still quoted in terms of bytes (8 bits) but it can be organised to have At each address 32 bits (4 bytes) or 64 bits (8 bytes)

There are 2 control lines associated with memory access
A **read** control line and
A **write** control line.
An active read means data is being read from memory to processor
An active write means data is written from processor to memory

Main memory contains data that can be interpreted either as an instruction
Or as operand data.
This interpretation is based on the principle of the fetch – execute cycle
Of an instruction cycle.

Instruction Cycle

1. Fetch Cycle
   (i) PC → MAR; Read (from memory to MBR)
   (ii) MBR → IR

2. Execute cycle
   (i) decode opcode of instruction
   (ii) update the PC register contents for the next instruction
   (iii) execute the instruction

Program flow

- A program has associated procedures or methods
- Branching to a procedure or method necessitates that present data and a return address be stored
- A part of main memory called the stack is used to remember data and addresses
- A stack pointer register in the processor is used to point at top of stack

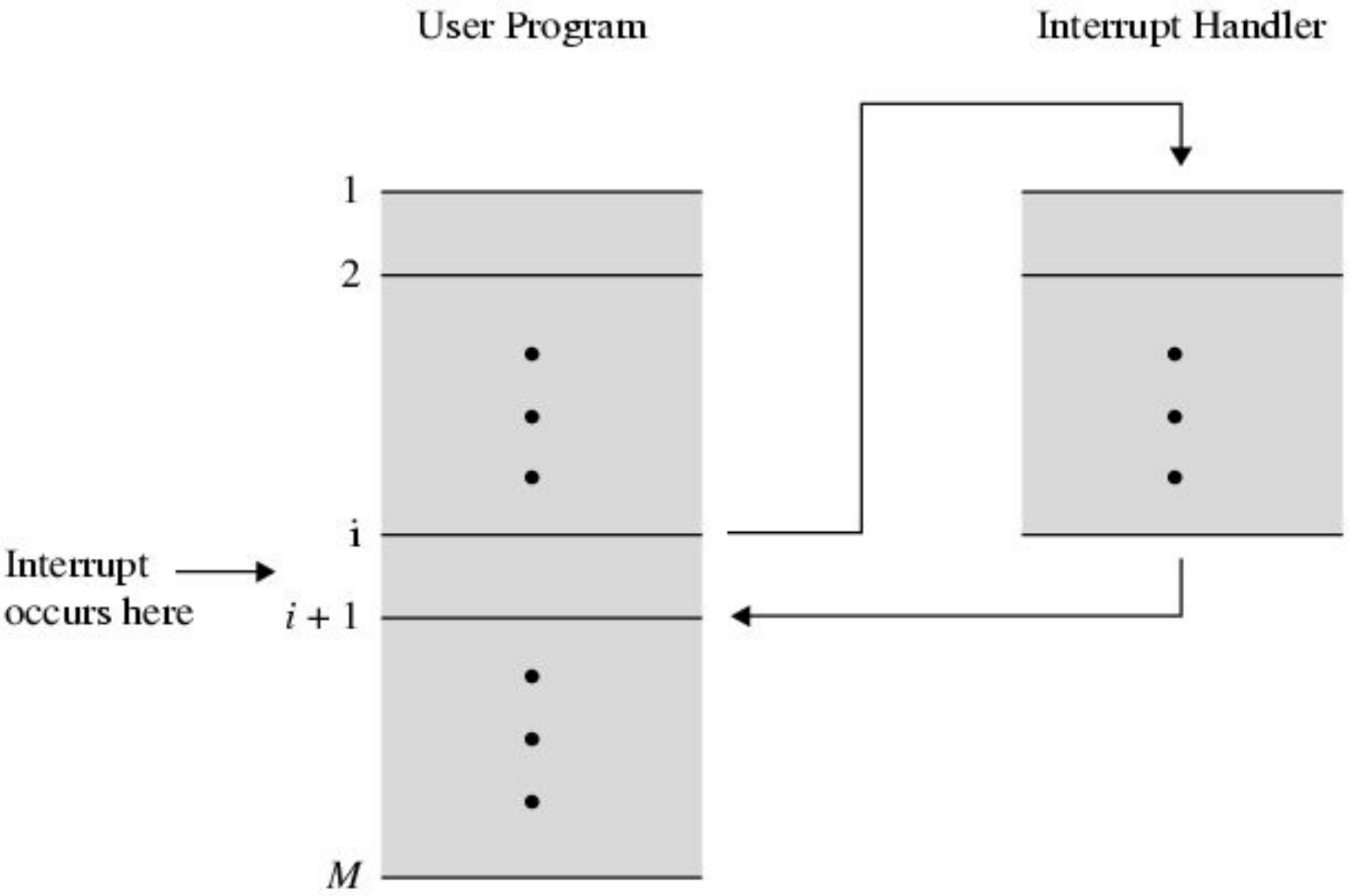
Interrupts

- 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

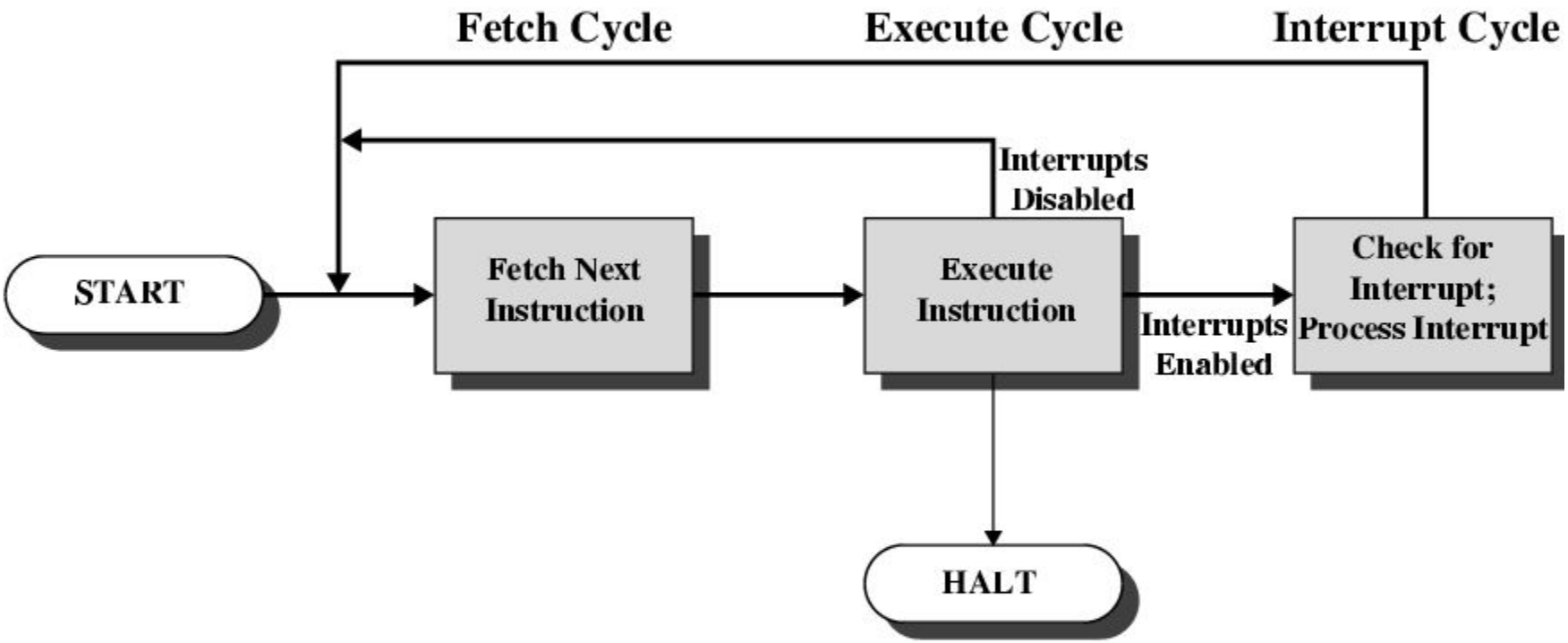
Interrupt Cycle

- 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



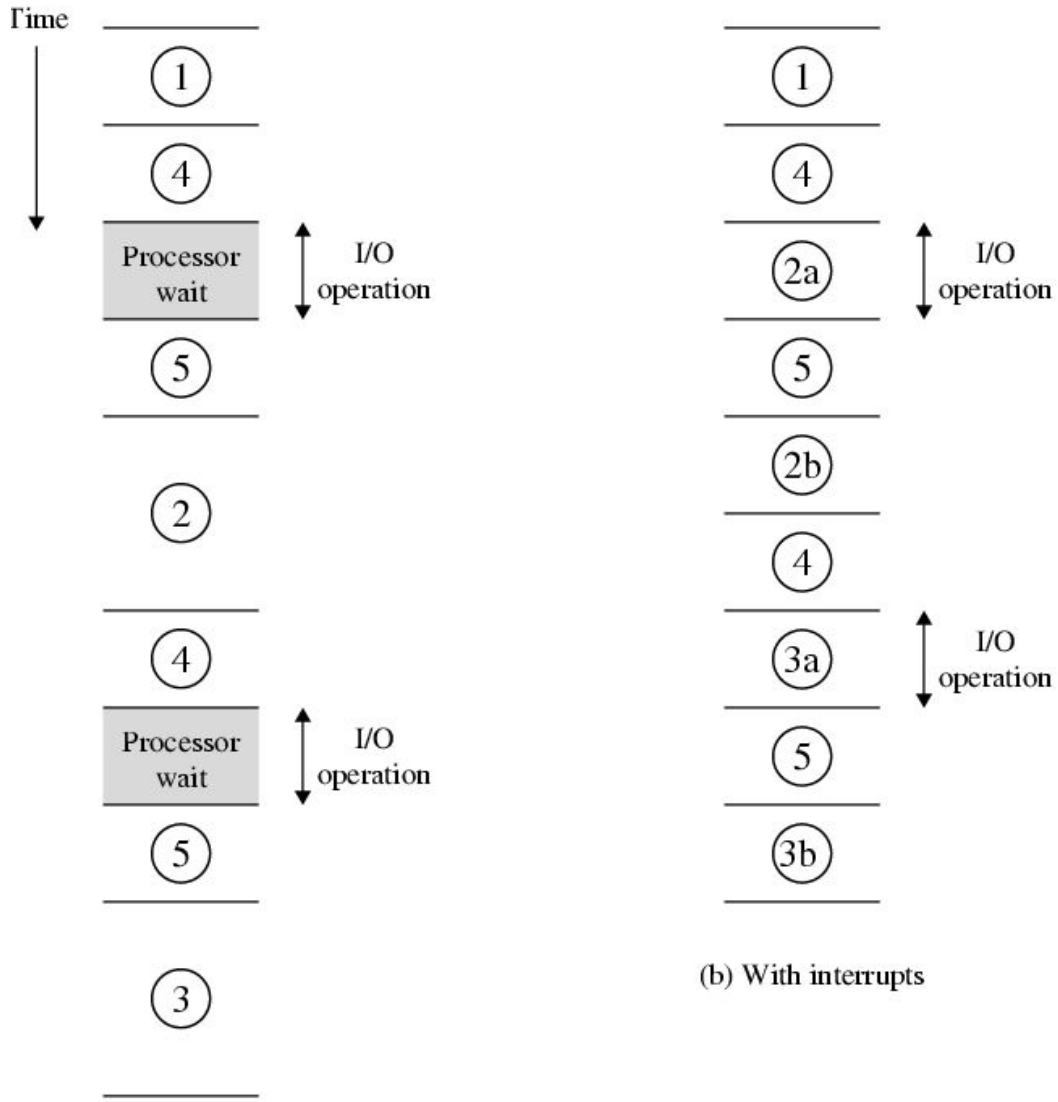
Advantages of Interrupts

- Processor initialises an I/O operation (4)(necessary both with and without interrupts)
- I/O Operation starts – without interrupts processor waits till end - with interrupts processor starts some other task (2a)

  At end of I/O - without interrupts processor finishes the I/O operation (5), then starts another task (2)

  with interrupts, processor receives an interrupt, stops the current task,(2a), services the interrupt to finish the I/O operation,(5), and continues the interrupted task, (2b).

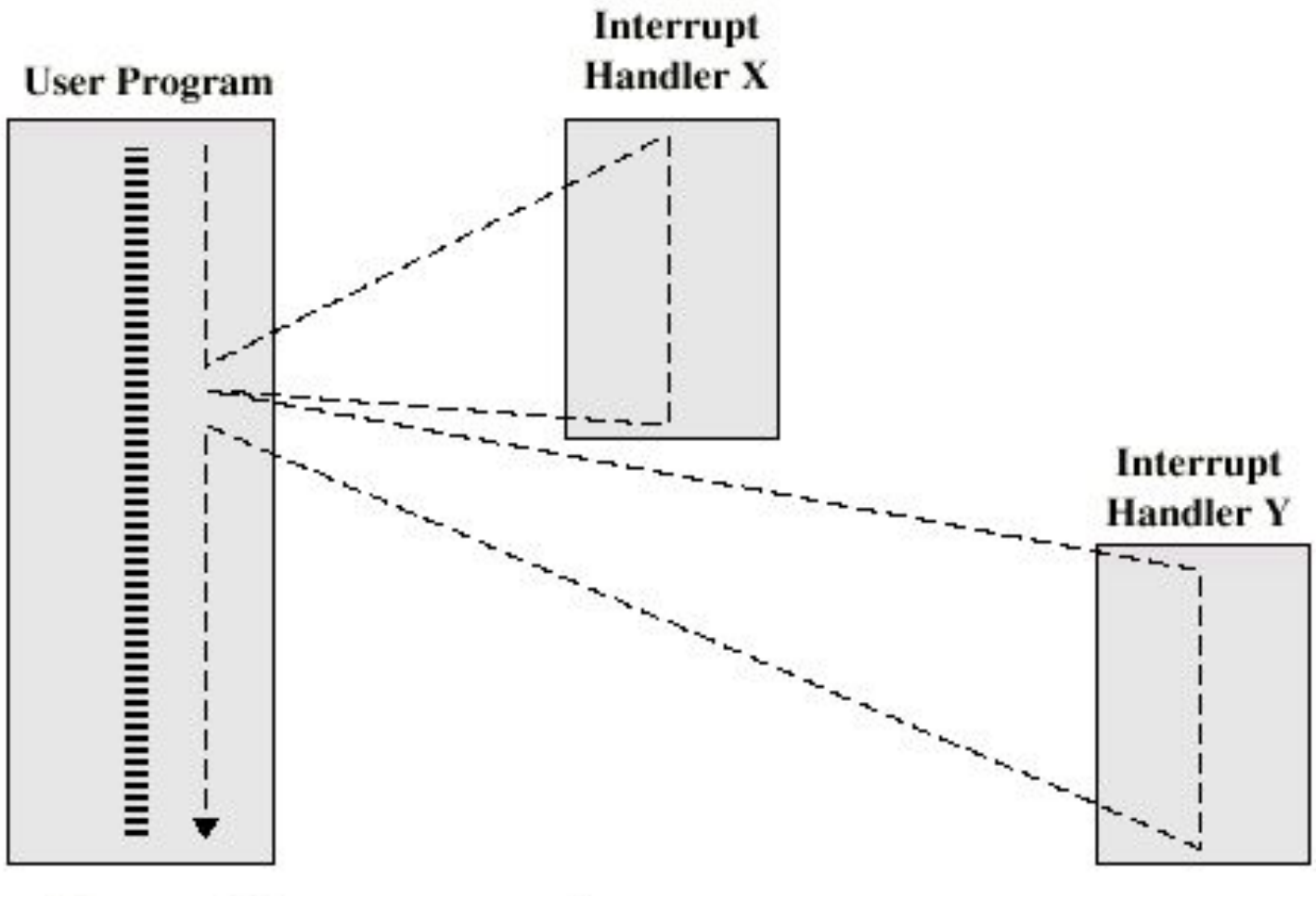
Program Timing - Short I/O Wait



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