EE2016 Microprocessor Theory & Lab, Fall, 2024

Week 9: Timing Diagrams and Performance Assessment

Dr. R. Manivasakan, OSWM Lab, EE Dept, IIT, Madras

Outline

- Opcode Execution and Timing Diagrams
  - Timing diagrams for simple opcodes
  - Cycles in microprocessor theory
- Performance Assessment
  - Total time taken for a program
  - Parallel Processing
    - Amdahl’s law
    - Gustafson’s law
-

Opcode Execution & Timing Diagrams

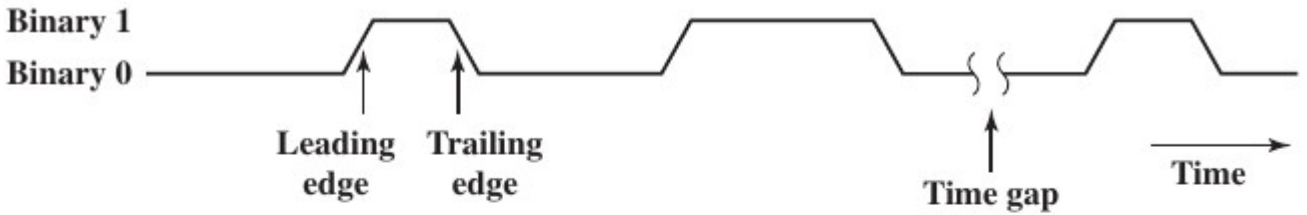
Organization

- Timing diagrams – Definition, conventions
  - Eg: timing diagrams for simple OPCODES
  - Cycles in Microprocessors
- Performance Assessment
  - Total average time taken for a program
  - MIPS
  - Amdahl’s law
  - Gustafson’s law

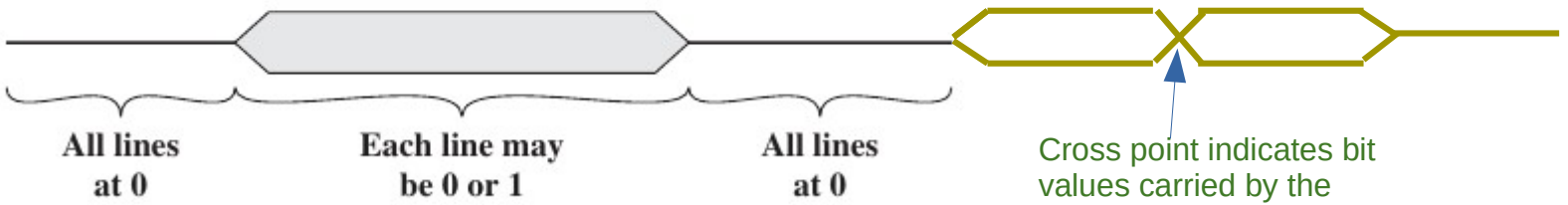
Timing diagrams

- Timing diagrams detail how an Opcode is being executed in the time axis
  - Defines how events (corresponding to an opcode) are coordinated on the three types of bus versus time
  - Allows one to
    - understand the actual performance of microprocessor, down to the time scale of clock interval (synchronous systems)
    - overall time an opcode takes in terms of clock cycles
  - One of the design aspects by the microprocessor / microcontroller manufacturer
- Timing diagram: Definitions and Conventions

Timing Diagrams: Definitions & Notations

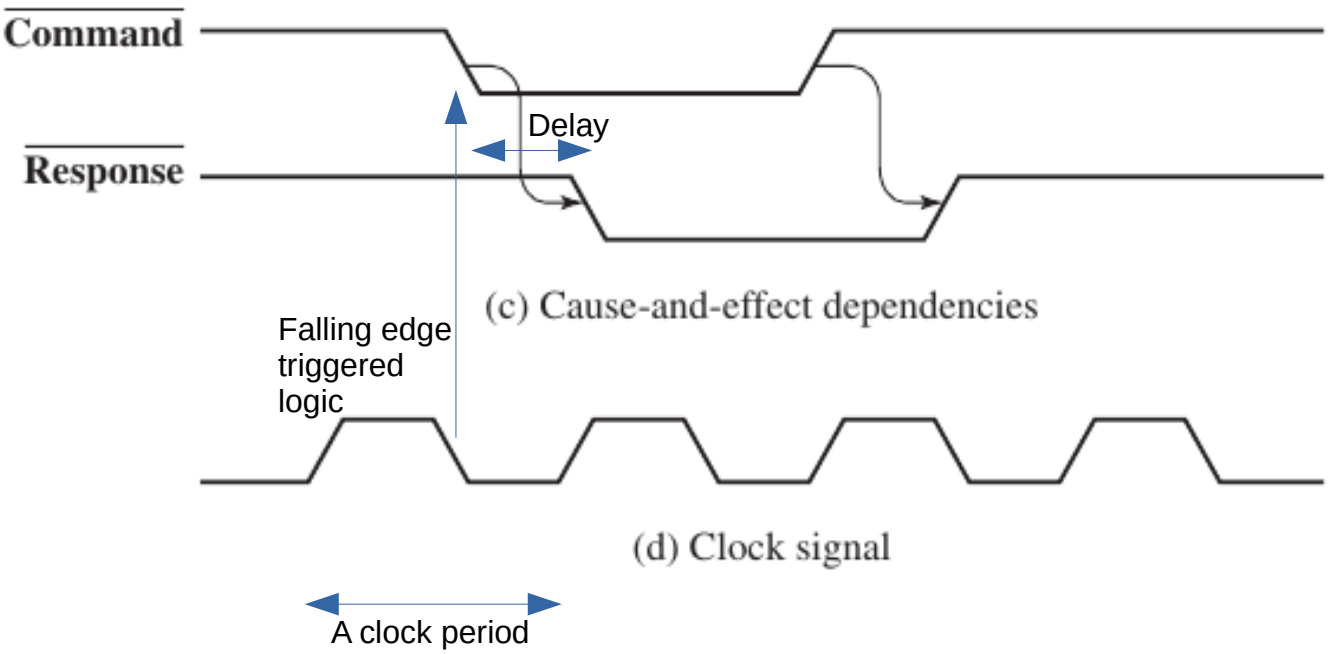


(a) Signal as a function of time



(b) Groups of lines

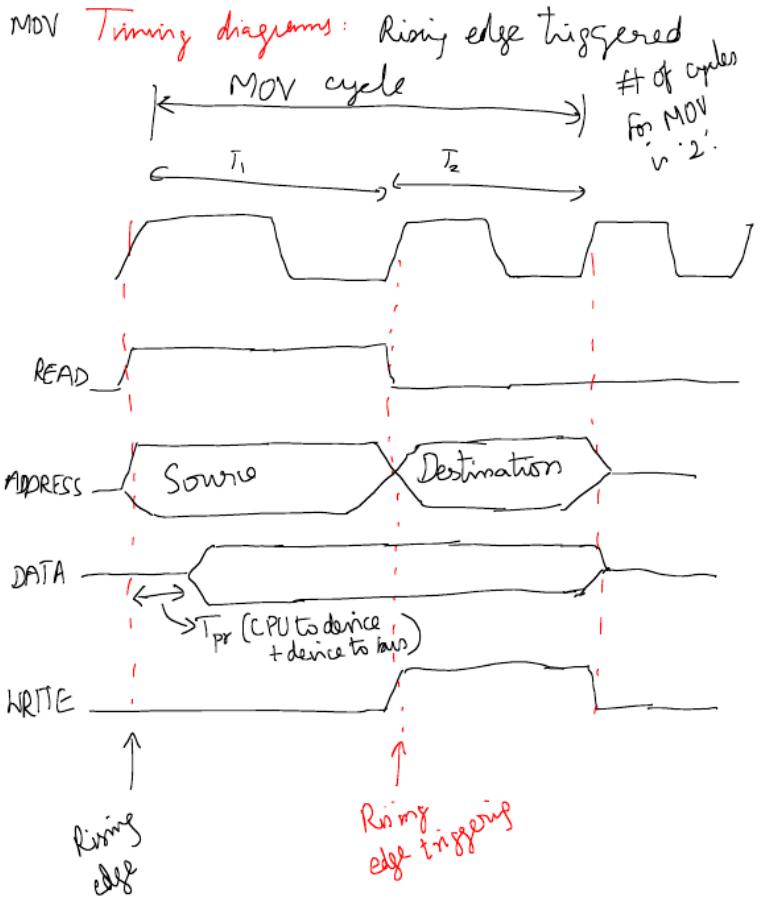
Timing Diagrams: Notations



Class on 21.10.2024

Timing Diagram for MOV Rd, Rs

- Step 1 (Cycle #1 – triggered at 1st rising edge of clk)
  - READ asserted
  - Source ADDRESS placed on address bus
  - Target source device responds after the round trip propagation delay (delay in reaching device + gate delay + propagation delay from device to bus).
- Step 2 (Cycle #2 – triggered at 2nd rising edge of clk)
  - Source ADDRESS on address bus removed
  - Target destination device ADDRESS placed
  - READ signal deactivated
  - WRITE signal asserted
  - Data written into the destination device
- Step 3 (End of Cycle #2)
  - Data line is reset (all lines = 0, DATA_BUS cleared)
  - WRITE signal deactivated (Control lines cleared)
  - Destination ADDRESS removed (Address lines cleared)

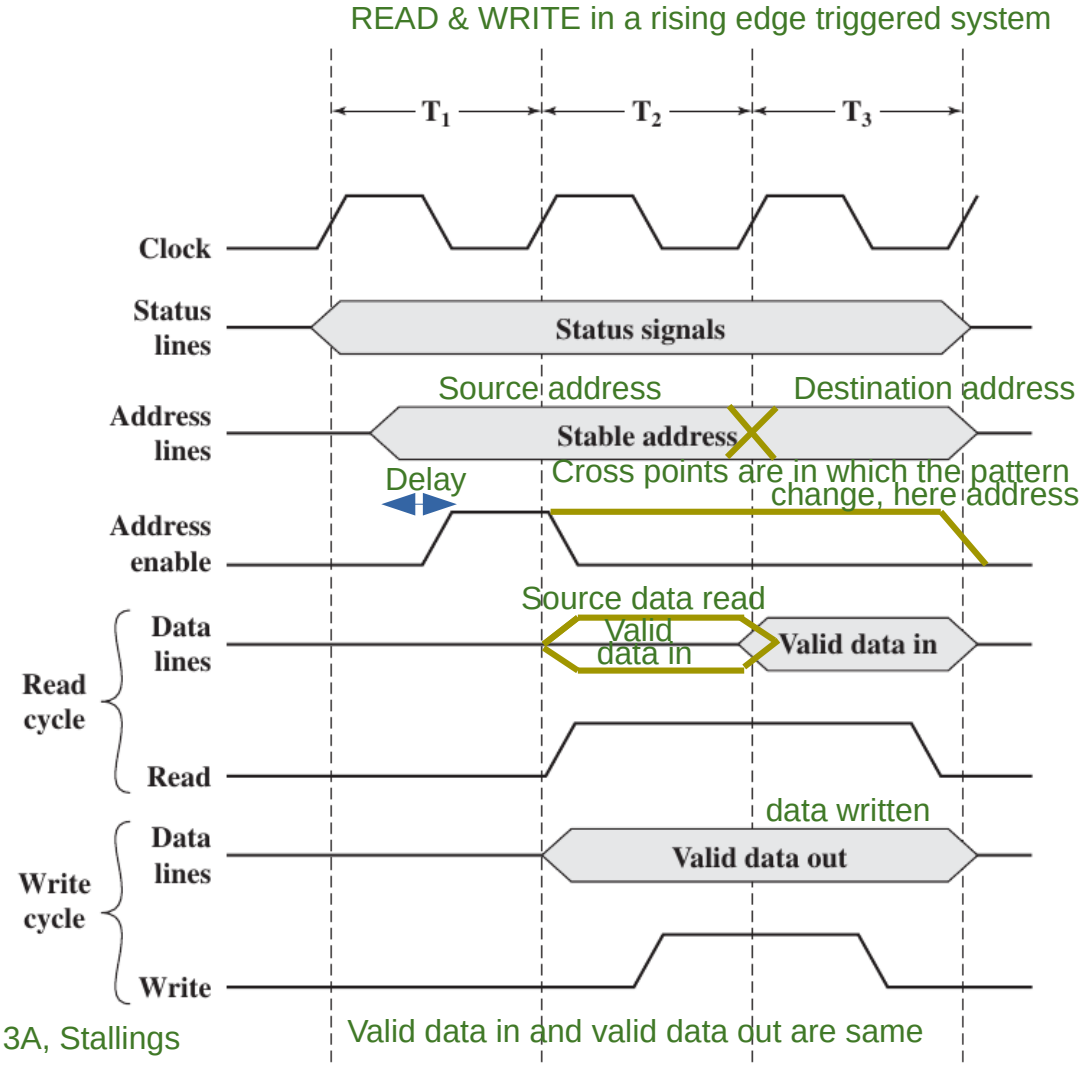


MOV Timing Analysis

- MOV takes single clock cycle in AVR (RISC based)
  - Microcontroller applications, Clk frequency is lower
- In CISC processor it takes 2 clock cycles
  - Clock cycle period is less --> higher frequency
  - --> microcomputer and microcontroller end up same absolute computational time
- Majority of AVR Instructions take single clock cycle
  - MOV, LDI, INC, IN, DEC, CPI (Compare with immediate), CP, COM, CLZ take single clock cycle
  - LPM takes 3 clock cycles

Timing in Synchronous Bus Operations

- All events (regarding digital signal transitions) are synchronous to the clock signal
  - Clock – essential component in a digital system
  - Clock period – defines clock cycle time or bus cycle time
  - Timing of events is dictated by the clock
  - Triggering happens at rising or falling edge of clock
- Read Process
  - First cycle
    - Address & Control (status) lines are asserted
    - 'Address enable' signal enabled
  - Second cycle
    - 'Read' signal asserted
  - Third cycle
    - Data available in data bus
    - Read signal drops
- Write Process
  - Second cycle
    - Valid data are asserted
    - 'Write' signal asserted
  - Third cycle
    - 'Write' signal deasserted

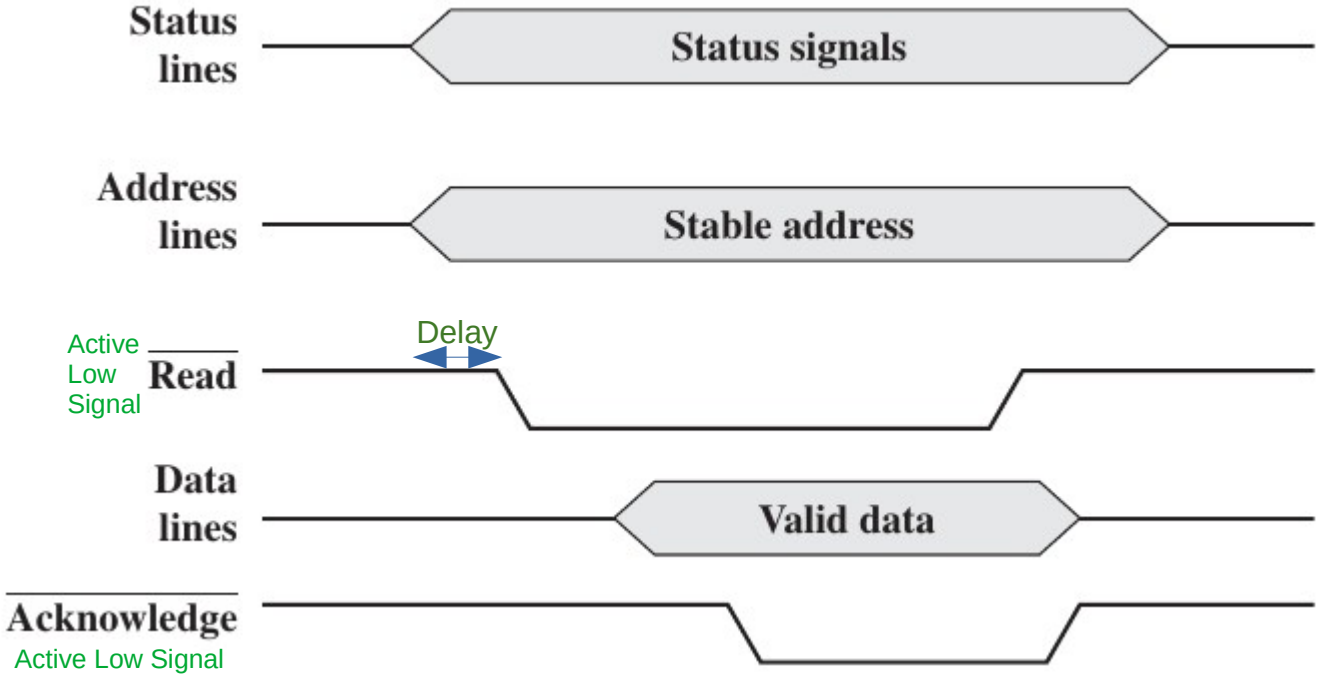


Ref: Chap 3, Appendix 3A, Stallings

Timing in Asynchronous Bus Operations

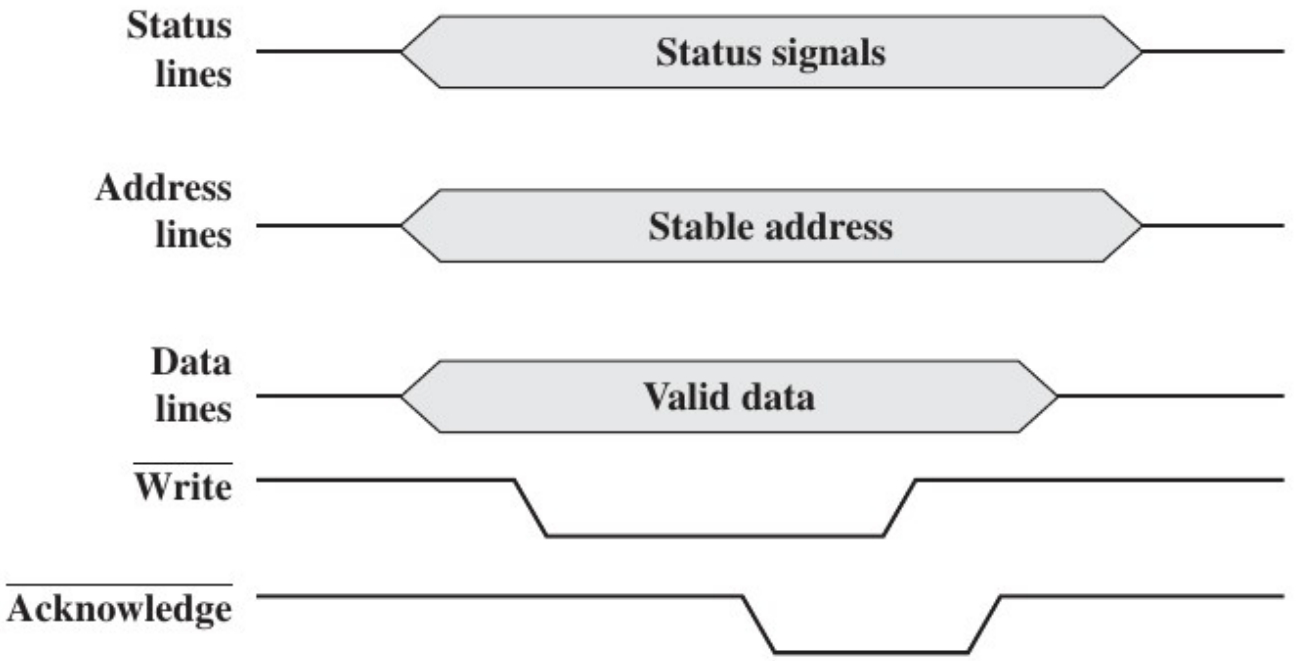
- All events (regarding digital signal transitions) are asynchronous
  - Occurrence of one event follows and depends on the occurrence of the previous event (after a fixed delay)
  - Signal transition is asynchronous
- Read Process
  - Address and status signals on the bus
  - Read Signal
    - Upon valid control and address lines
  - Memory
    - Decodes address and places the contents of addressed memory location into the data line
    - Acknowledge signal is asserted
- Master read
  - Once the master reads data
    - It deasserts the read signal
  - Memory
    - deasserts the data line
    - Deasserts acknowledge signal

Asynchronous Bus Operations: READ



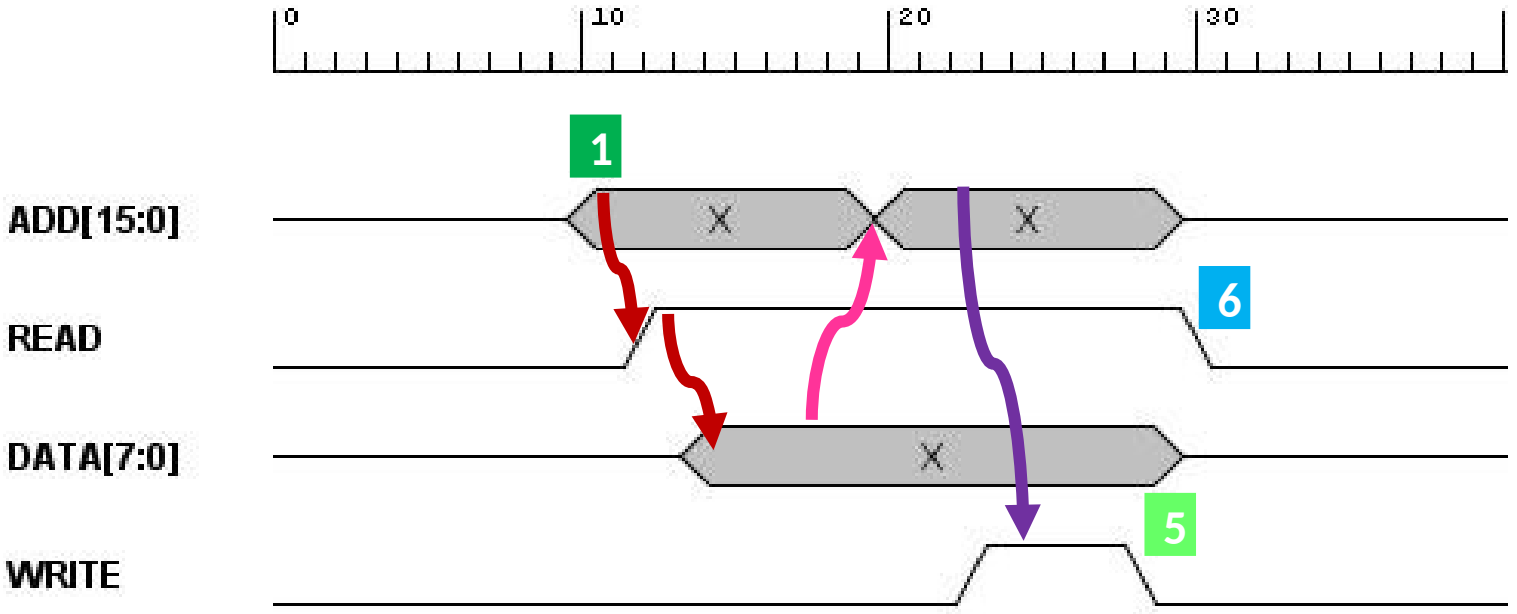
Timing in Asynchronous Bus Operations: Write Operation

- Write Process
  - Master (CPU) asserts address and status signals on the bus
  - Also places data on data lines
  - Memory module responds by
    - Issuing ‘write’ command – copying data from data bus into memory location
    - Acknowledge signal is asserted
  - Master
    - Drops write signal
    - Memory module drops acknowledge signal



Dissecting the MOV operation

- With 8 bit source (R1) and destination (R2) registers, 8 bit data bus, 16 bit Address bus



1. Set the source address into address bus
2. Assert Read signal so that data becomes available on the data bus
3. Set destination address in the address bus
4. Assert Write
5. Remove write
6. Remove read signal
7. Remove address

Synchronous Vs Asynchronous

- Synchronous Bus Operation
  - is simpler to implement & test
  - Less flexible than asynchronous timing
  - System cant take advantages of advances in device performance
- Asynchronous Bus Operation
  - Devices a mixer of slow or faster using older or newer technologies are allowed to share the common bus

Concept of Cycles in Microprocessors

- Cycles
  - Bus cycle
  - Clock cycle
  - Memory cycle
  - Machine cycle
  - Instruction cycle
  - Processor (or CPU) cycle
- Clock cycle and bus cycle are same
- Memory (access) time (or cycle)
  - Time for READ from a memory location or WRITE into a memory location

Concept of Cycles in Microprocessors

- Memory cycle time
  - Time between two consecutive memory actions (READs, WRITEs, READ – WRITE, WRITE-READ)
- Memory cycle time is slightly larger than the memory access time.
- Machine cycle
  - Time for Fetch-Decode-Execute
- Instruction cycle and Machine cycle are same
- CPU cycle is the time spent by CPU in decode – execute part of instruction cycle. The decode – execute part of a single instruction might consist of many CPU cycles.
-

Performance Assessment

- Criteria for choosing a muC for an application
  - Its availability in the market
  - Cost
  - Size (foot print)
  - Ease of design and development
  - Time to market
  - Reliability
  - Power consumption
- All operations (events) within the microprocessor are synchronized with the pulse of the clock (raising edge or falling edge) – MuC based on synchronous bus operation.
- The number of cycles required per instruction can vary from one clock cycle time to several clock cycles.

Performance Assessment

$$NCPI_{avg} = \sum_{i=1}^{n} \frac{NCPI_i \times NI_i}{TNI}$$

where No of cycles/instruction of type 'i' - $NCPI_i$

No of instructions NI_i of type 'i'

TNI - Total # of instructions

T_{net} - Net time taken for execution of the above program

$$= (TNI)(NCPI_{av}) T_{clk} \quad \text{— (1)}$$

Performance Assessment

- p - Number of CPU operations for decode & execute part of a given instruction
- m – number of memory cycles for memory access operations corres-ponding to a given instruction

$T_{CPU} = N_{CPU} T_{clk}$

$T_{Mmry} = N_{mmry} T_{clk}$

Average # of cycles per instruction $NCPI_{avg}$

p - # of CPU cycles for decode & execute

m - # of memory cycles for memory ref / instruction

$$NCPI_{avg} = p N_{CPU} + m N_{mmry}$$

$$= N_{CPU}\left(p + m\frac{N_{mmry}}{N_{CPU}}\right) = N_{CPU}(p + ma) \quad (2)$$

Typically $a \gg 1$

N_CPU is the number of clk cycle per CPU operation

N_Mmry is the number of clk cycle per Mmry operation

Net time spent in a program

$$T_{CPU} = N_{CPU} T_{clk}$$

$$T_{Mmry} = N_{mmry} T_{clk}$$

Typically $a \gg 1$

Plugging in (1), the eqn. (2),

$$T^{net} = TNI\ N_{CPU} (p + ma) T_{clk}$$

Class on 22.10.2024

Measure of computer's speed

- Million Instructions per Second (MIPS)
  - MIPS = f/(NCPIav * 10^6)
- Kilo Instructions per Second (KIPS)
- Billion (Giga) Instructions per Second (GIBS)
- Mflops –
  - Mega or Million floating point operations per second

Effect of Performance Measures on Design Parameters

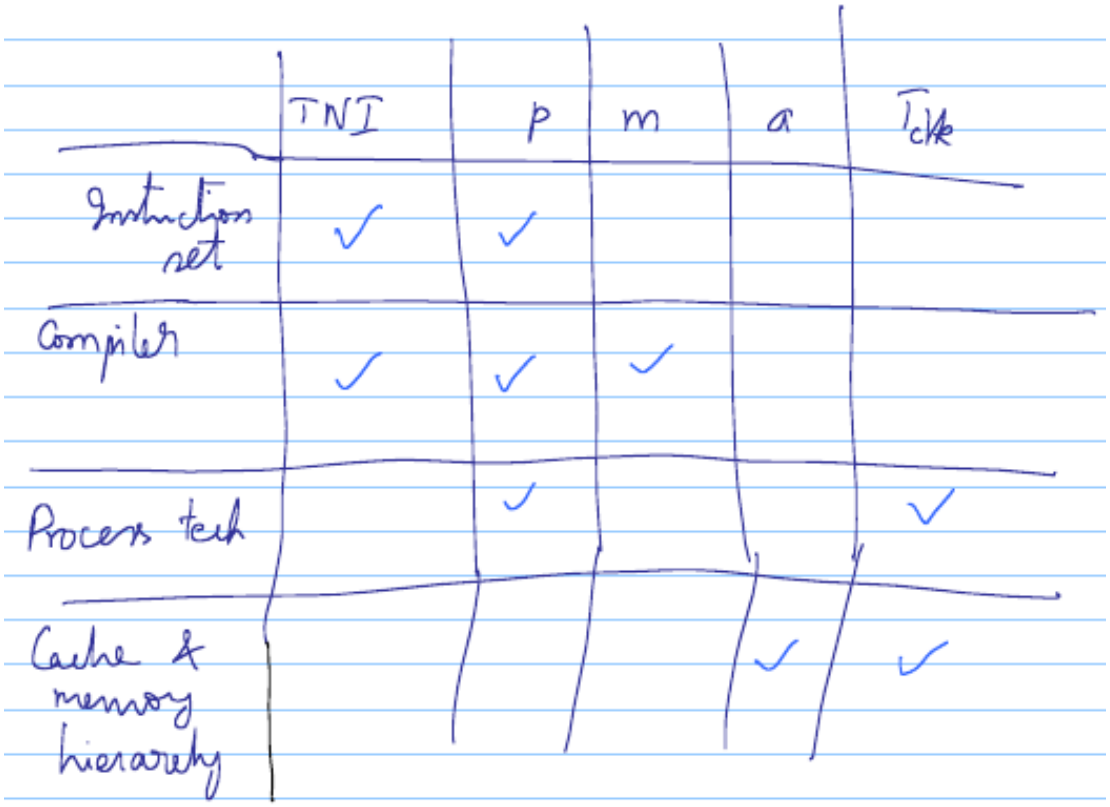
$$T_{CPU} = N_{CPU} T_{clk}$$

$$T_{Mmry} = N_{mmry} T_{clk}$$

Compiler here refers to the C compiler (or any higher level language) and hence higher level Syntax / commands are converted into machine code & thus compiler affects the TNI. So also p & m

Process technology affects p?

Cache & memory technology affects m



Performance Assessment: Parallel Processing Case

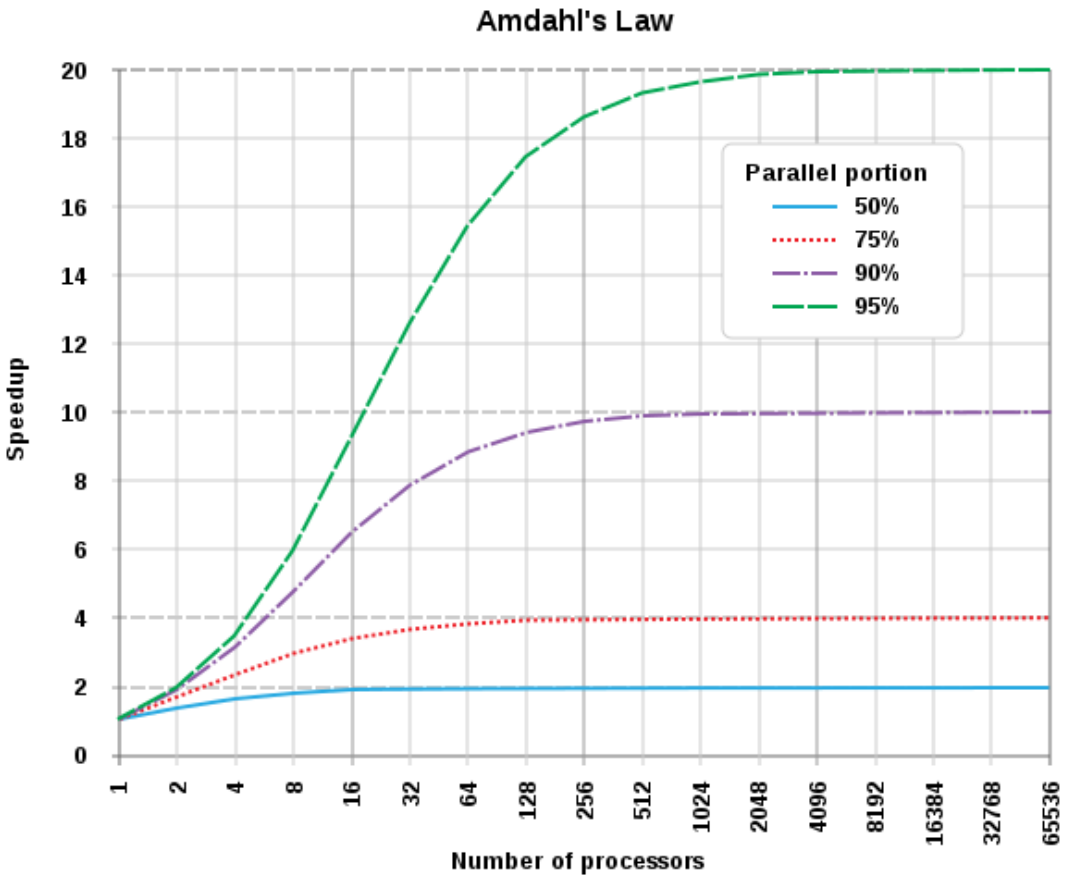
- Multicore processors
  - Parallel processing
- Concept of parallel processing
  - Eg: Matrix addition
  - Component of a program which could be parallelised
- Parallel processing - levels
  - In bit level
  - In instruction level
  - In computation level
  - In task execution level
- S

Performance Assessment: Amdahl’s Law

- Given a program (with work W) in which a fraction ‘a’ could be parallelised, then time taken by
  - Single processor: Time taken = (1-a)T+ aT = T
  - N ~ Multi processor: Time taken T(a,N,W) = (1-a)T + (aT)/N
- Speedup factor S = (TW)/[W T(a,N,W)] = 1/[(1-a) + a/N]
- Amdahl's law assumes that the entire problem is of fixed size W, so that the total amount of work to be done in parallel is also independent of the number of processors

Performance Assessment: Amdahl's Law

- Speed-up increases with the increase in parallelisable factor
- Limit when the resources tend to infinity



Performance Assessment: Gustafs

- In practice, as more computing resources become available, they tend to get use (larger datasets), and the time spent in the parallelizable part often grows much inherently serial work.
- In this case, Gustafson's law gives a less pessimistic and more realistic assessmer performance
- Parallisable part of work load = a
- Let the work load be W for the single processor case. Then
  - W(a, N, T) = (1-a)W + aW
- Given fixed time T of execution, the renegotiated work load it can take (due to N is given by W(a, N, T) = (1-a)W + NaW
- Gustafson's law assumes that the total amount of work to be done in parallel inc the number of processors N (for T fixed).
- Speedup factor S = [T*W(a,N,T)]/(T*W) = W(a,N,T)/W = 1-a + N*a gives theoretic of the execution of whole task at fixed time T

Performance Assessment: Gustafson’s Law

- Gustafson’s law
-