Computer Organization
(Processor Level Design)

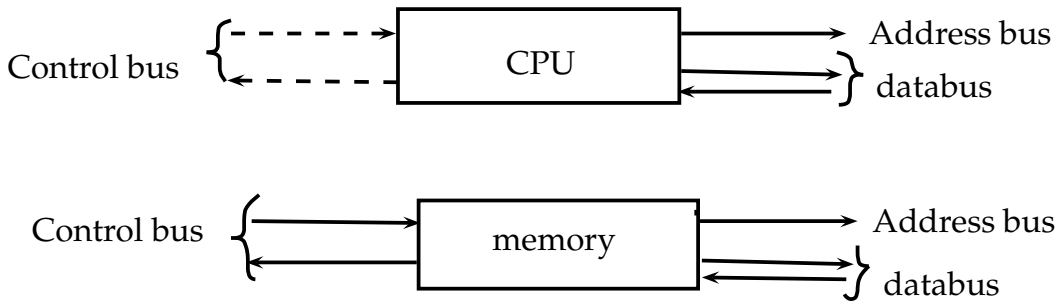
KR Chowdhary
Professor & Head
Email: kr.chowdhary@gmail.com
webpage: krchowdhary.com

Department of Computer Science and Engineering
MBM Engineering College, Jodhpur

November 14, 2013

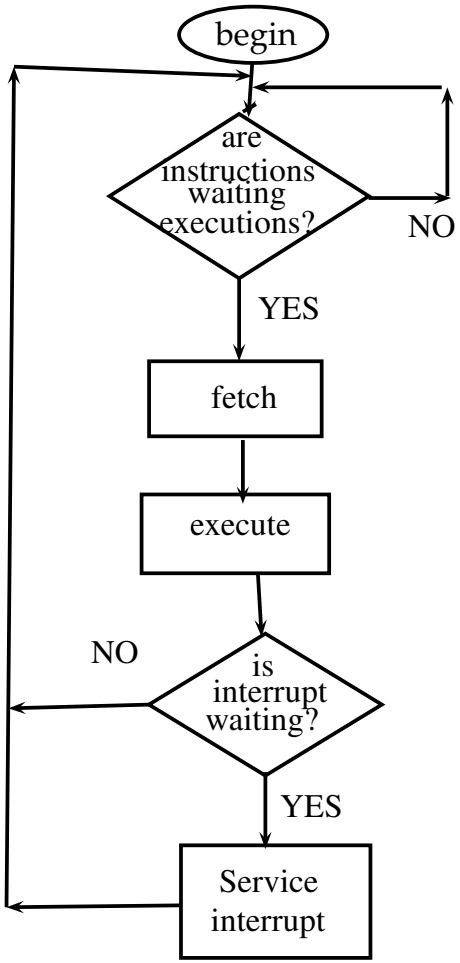
Processor level design

- CPU or processor (instruction sets, their execution times, program control unit, CPU's communication with external devices)
- Memories (different technologies, varying cost/performance). 1) Main memory: fast, small, controlled by CPU, 2)secondary: slow, large, inexpensive, communicate via main memory to CPU. Serial/parallel access.



- Input/output Devices(communicate with outside world, data transducers, can be directly controlled by cpu, IOPs for IO devices)
- Interconnection networks (word level buses, often shared (why?), priorities, asynchronous communication with CPU (why?), handshaking communication, bus control by cpu/IOP). Information transferred is in words, Interconn. networks (busses) provide dynamic connection between components, handshake, synchronous/clocked

Instruction cycle

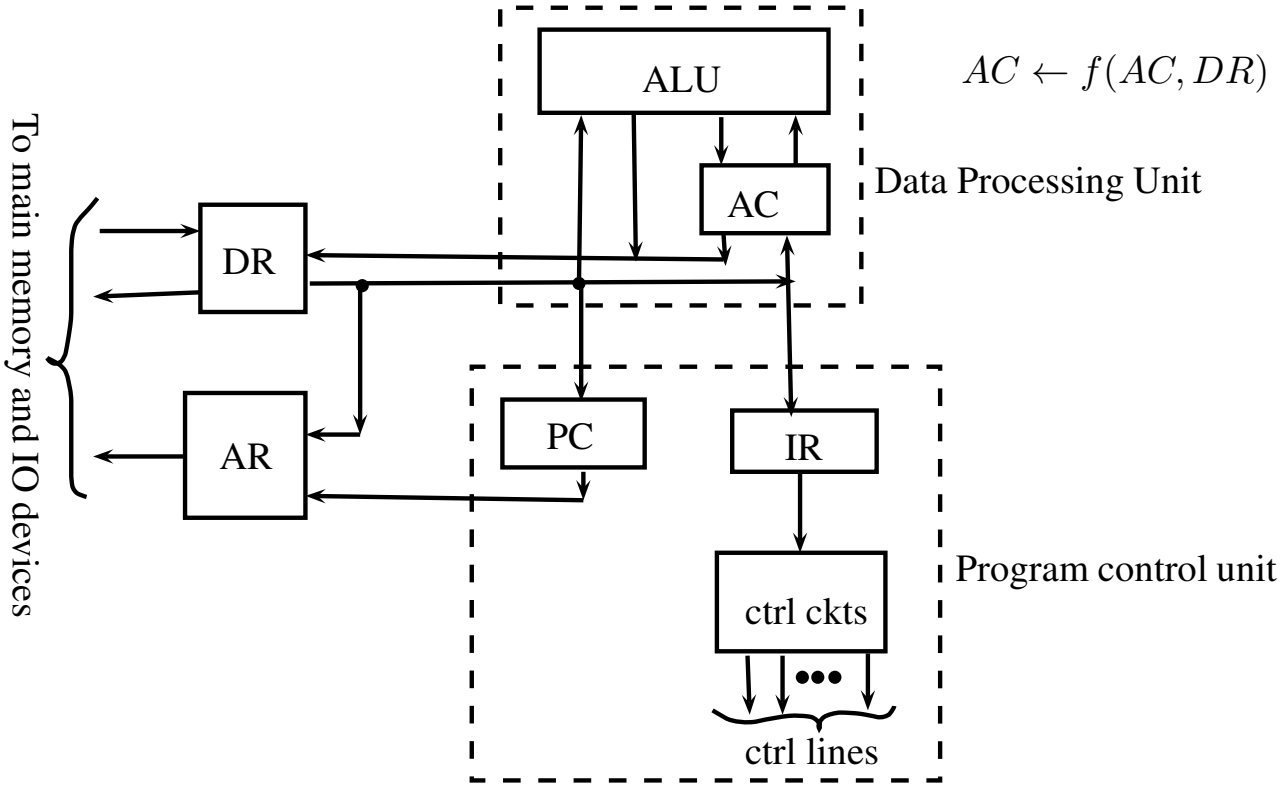


- Instruction cycle=fetch cycle + execute cycle. $\Rightarrow$ sequence of micro-operations
- t_{cpu} = micro-operation time = CPU cycle = CPU clock rate
- CPU Interrupts: Request to CPU by IO devices.

  Interrupt handling program, how to test the presence of interrupt? t_M/t_{cpu}?
- Memory Mapped IO: IO addresses are like memory locations.
- IO mapped IO or Isolated IO:

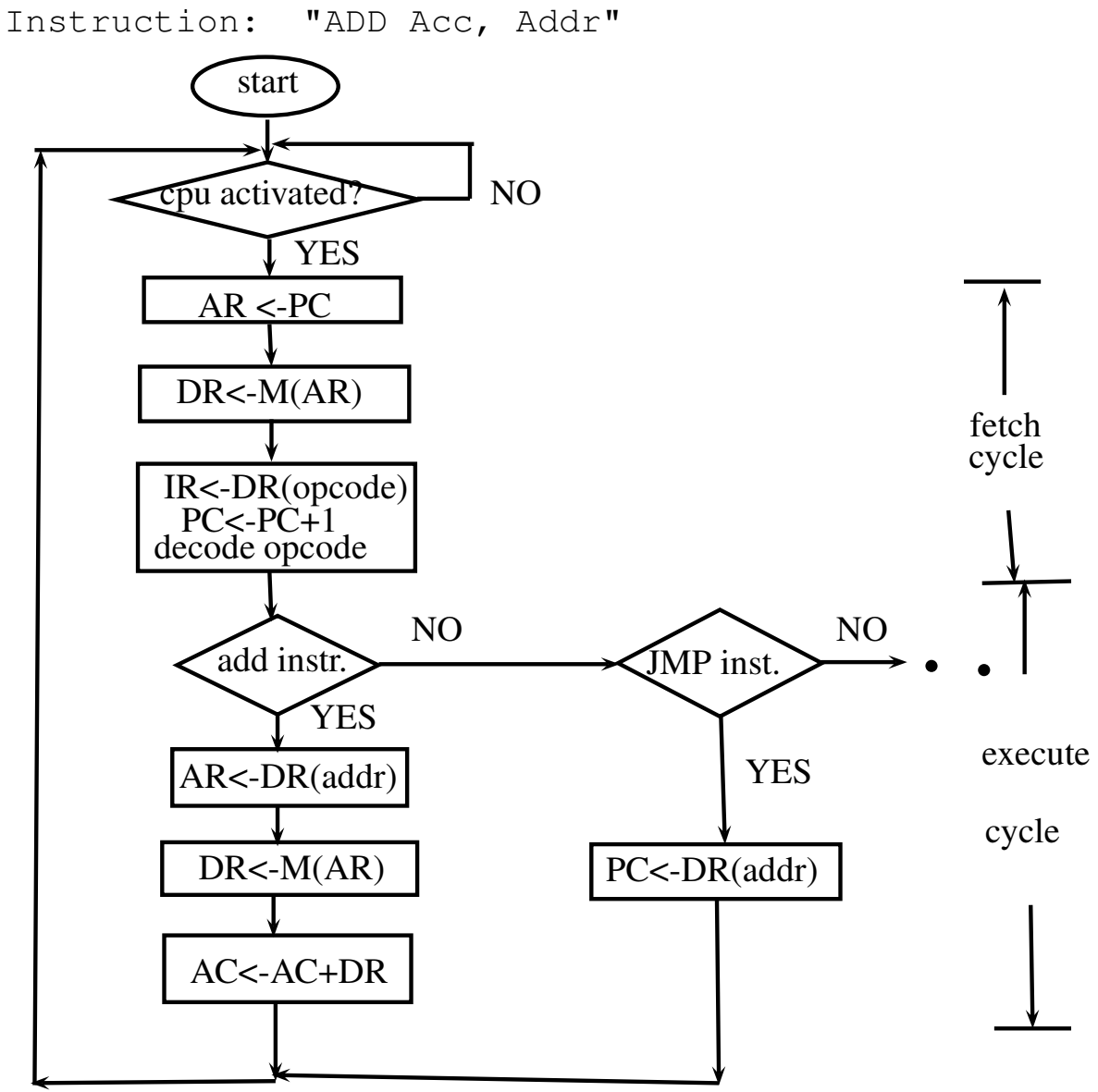
  Only data movement takes place with IO.

CPU Architecture



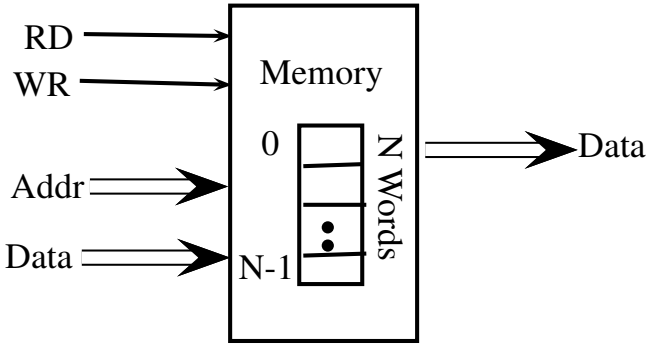
- One CPU Cycle, t_{cpu} = smallest micro-operation of CPU, $1/t_{cpu}$ = maximum clock freq. of CPU. Execution time = no. of CPU clock cycles $\times t_{cpu}$
- One memory cycle t_m = time spent between address applied to memory and data released by memory.
- $t_m/t_{cpu} \approx 10$.

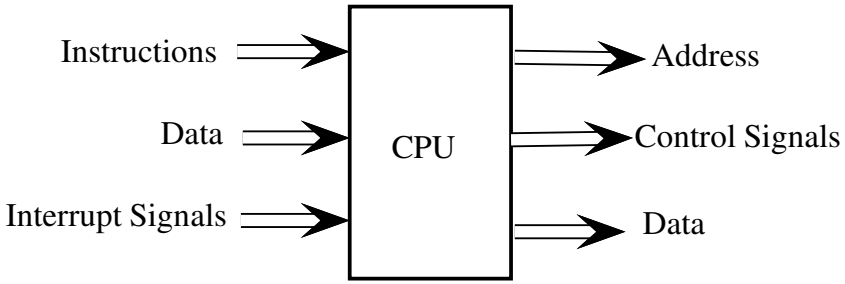
Instruction Cycle Example

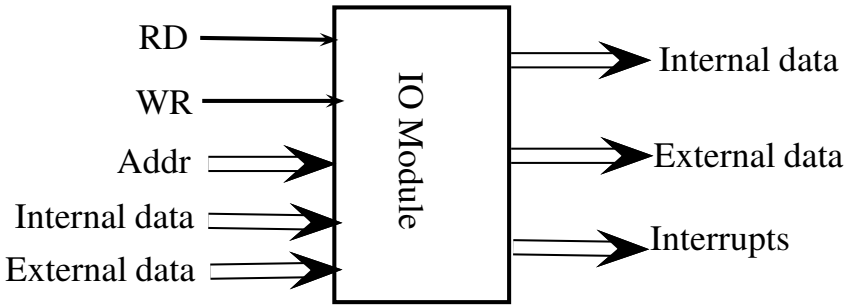


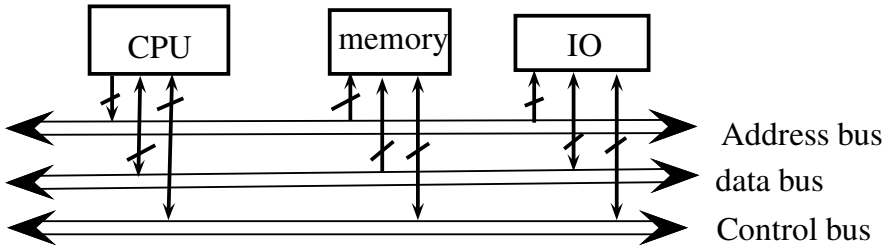
Interconnect structures

- These provide the connection path for connecting CPU, Memory, IO.
- Control signals are: MRD, MWR, IORD, IOWR, Transfer ACK, BUS Req, BUS Grant, INTr req., Intr ACK, Clock, Reset

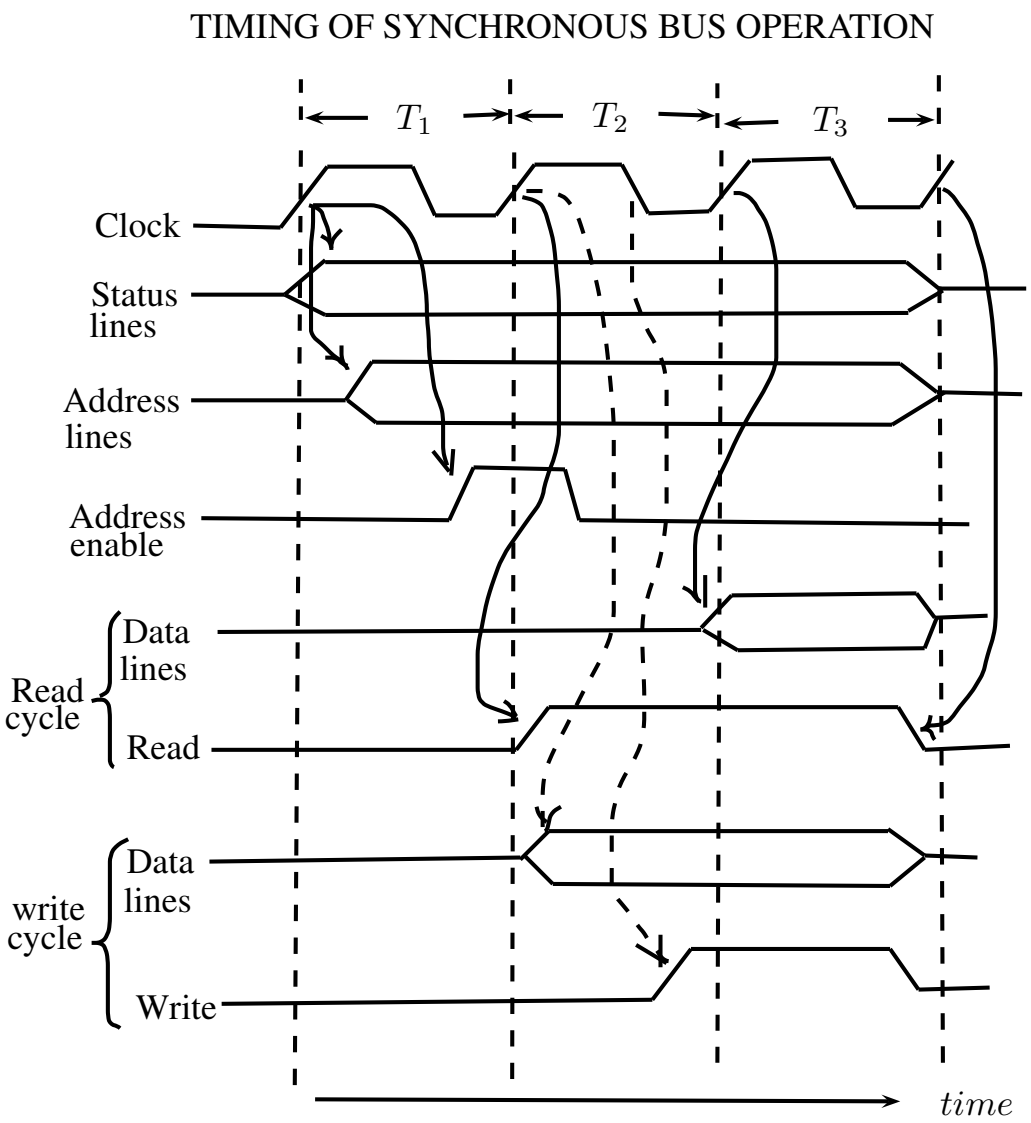






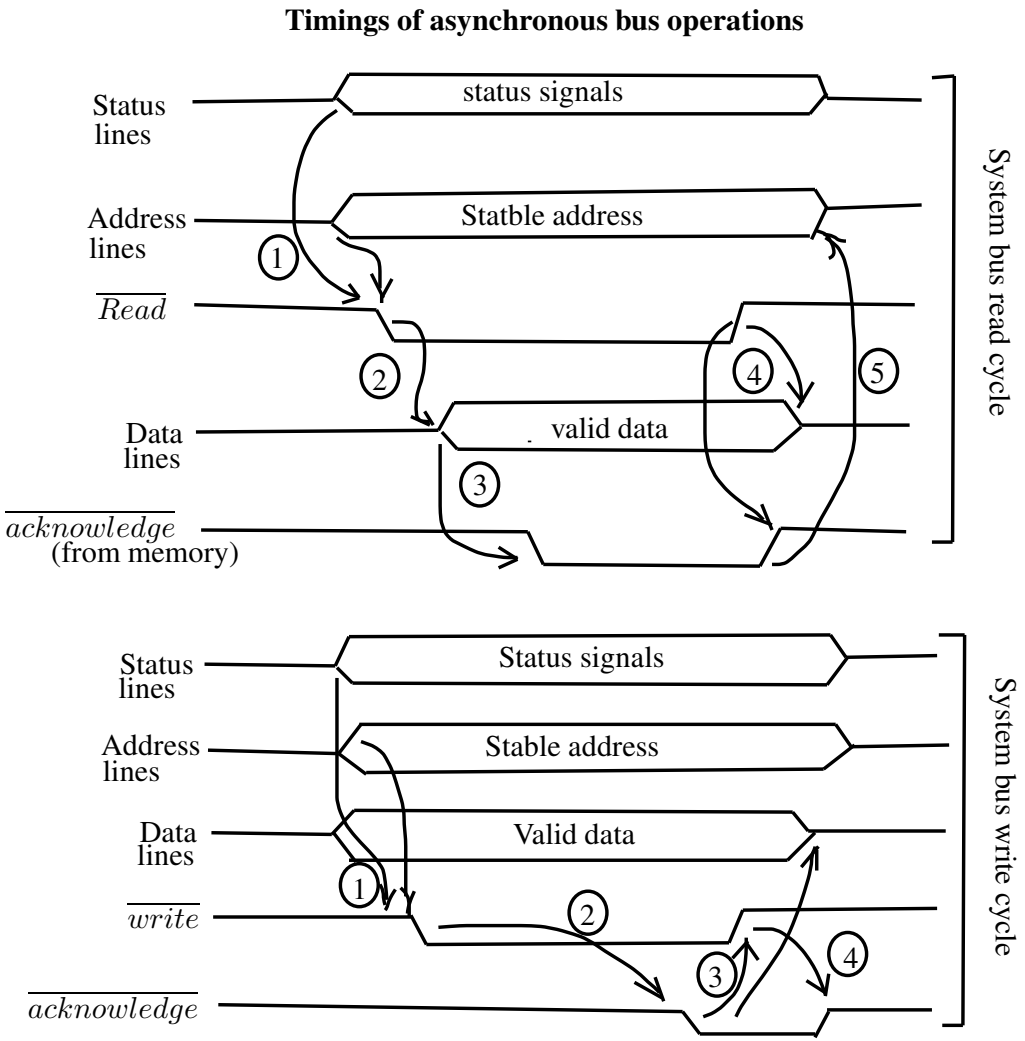


Timings of Synchronous Bus operations



- Waveforms show the memory Read/Write cycles.
- Occurrence of events is determined by a clock. clock cycle = bus cycle.
- all devices on the bus read the clock line, all events start at begin of clock cycle
- Synchronous: timing of any transition is known in advance. Asynchronous: depends on the availability data and readiness of devices to initiate bus transition.

Timings of Asynchronous Bus operations



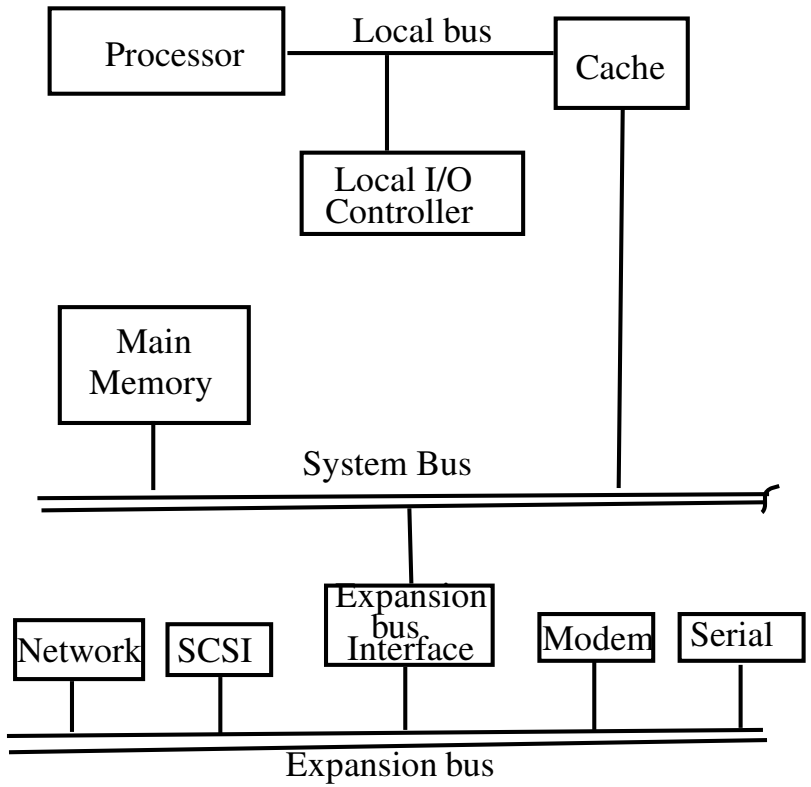
- Occurrence of events is not determined by a clock.
- Occurrence of one event on bus follows the other.
- The CPU is master for data transfer. It can be IOP also.
- Synchronous is simple, but tied to clock (less flexible). Thus, high performance device cannot contribute.
- Syn: advance features of devices can contribute, mix of slow and fast devices can work together.

Buses

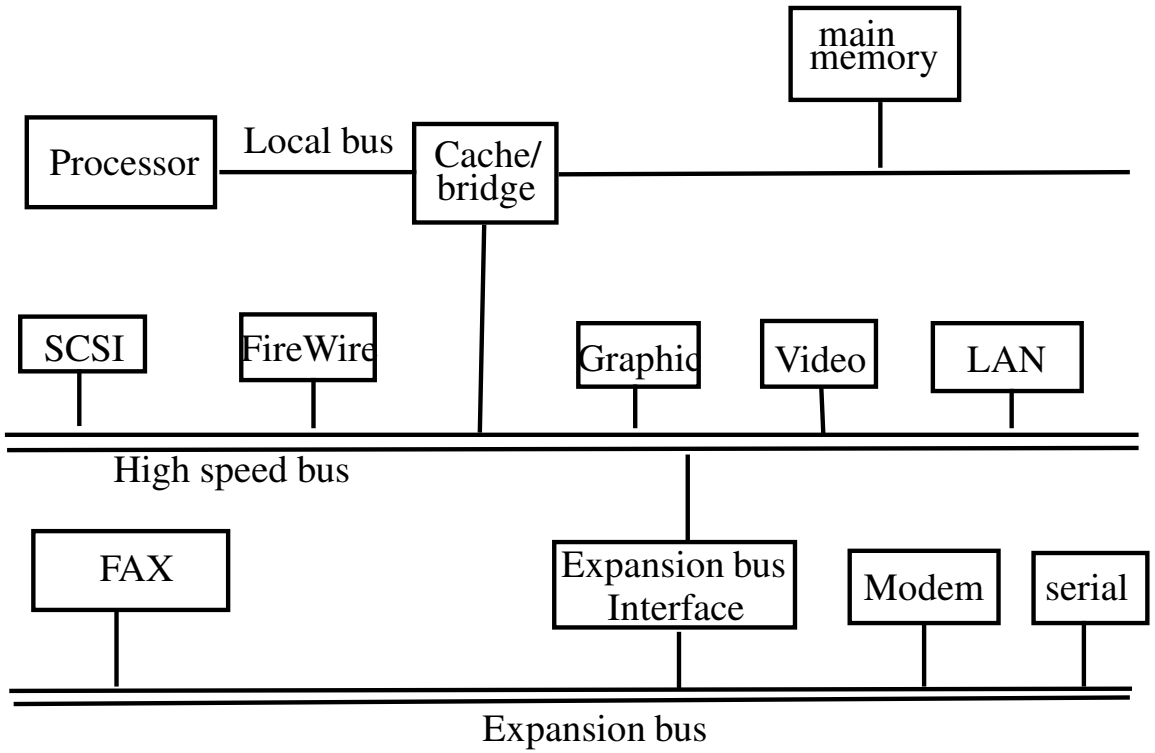
Buses types

- Inside CPU (CPU Bus or Onchip Bus:), to connect registers, ALU, and cache.
- System bus or Onboard bus: Between Processor and memory
- Peripheral bus: To connect fast peripherals, like graphics card, LAN adapter, with the main memory for high speed data transfer as well as to connect slow devices. Connects from dual ported memory.
- More devices requires longer bus, results to propagation delays, coordination problems between devices.
- More devices cause bottleneck for data transfer, forces to design wider buses(32 to 64 bits)
- Or more than one buses

Traditional Bus structure:



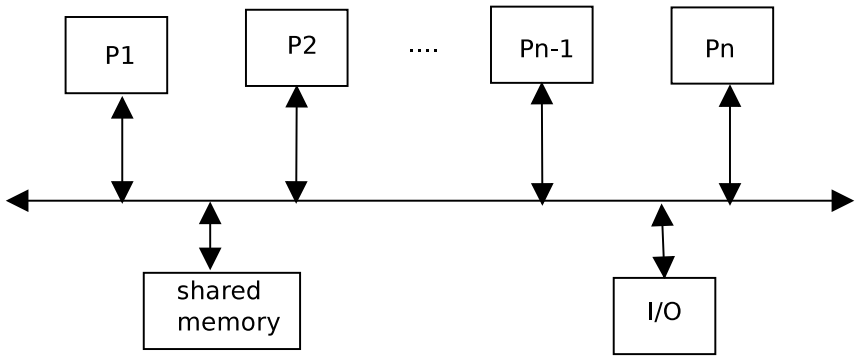
High performance Bus architecture



Bus Types:

- Dedicated: separate address, data buses (common arch.)
- Multiplexed: first address is sent, after some time data is sent. (time multiplexed)(adv./disadv.?)

Multiprocessor bus



Single bus system

- Bus contention arises when more than one processors competes to access the bus.
- In Single bus system bus arbitration is required to resolve the contention.
- The processor that wants to use the bus submits a request to "arbitration logic".
- Arbitration logic decides based on some priority, which processor should be granted the bus access during a certain period (bus master).
- Passing bus mastership is through handshaking: 1) bust request, 2) bus grant.

Bus Arbiter

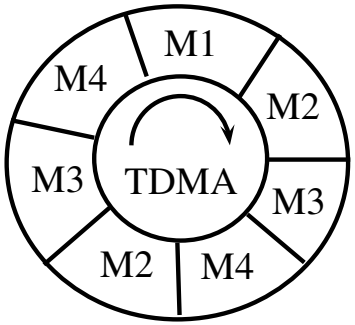
The arbiter samples the request on rising edge of clock; the predefined algorithm decides as which master is next to gain access to the bus.

Static fixed priority Algorithm:

1. Each master is assigned a fixed priority value,
2. Master of highest priority is granted bus
3. Adv: simple, inexpensive
4. Disadv: In high traffic, starvation to low priority masters.

Round-robin Algorithm:

1. Keeps track of last master
2. Starts from next master
3. Adv: easy to implement
4. Disadv: privileged masters are benefited



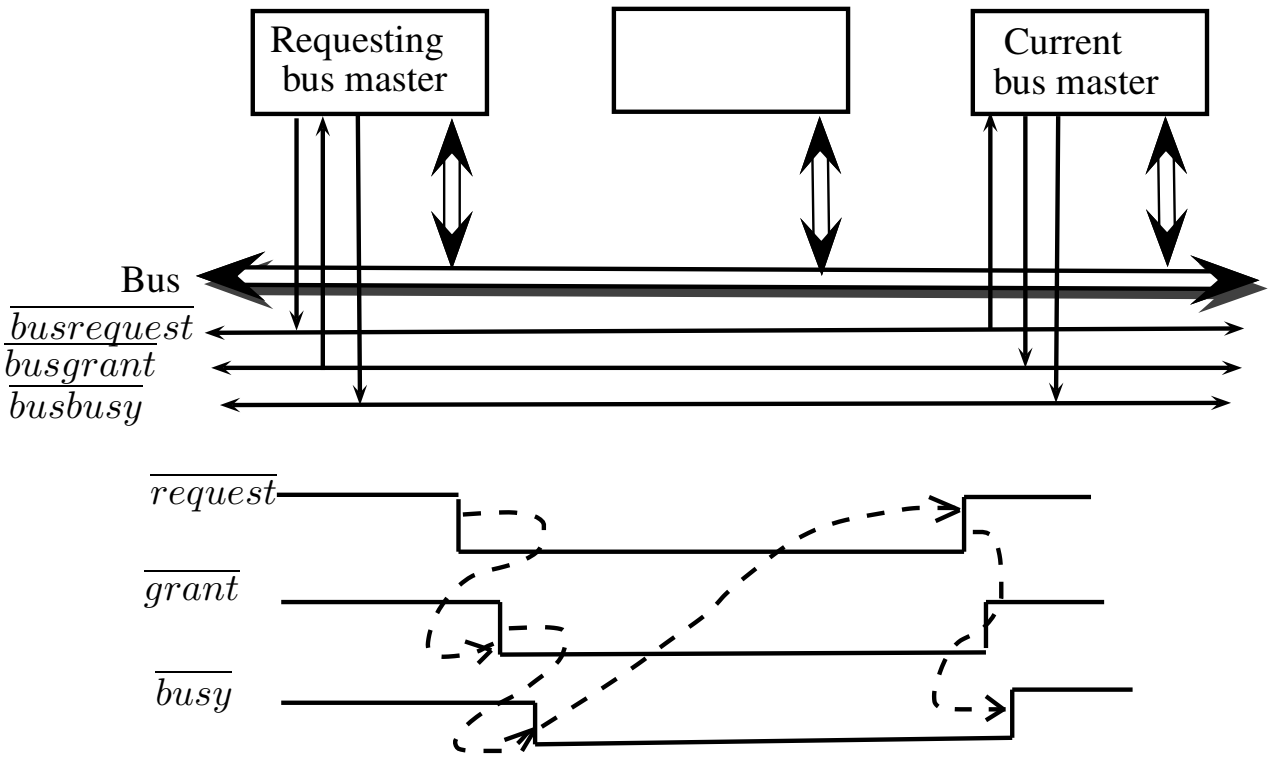
Order for Masters:
M1, M2, M3, M4, M2, M3, M4

Bus arbiter

Other algorithms:

1. Random priority algorithm
2. Equal priority: when two or more requests are made, there is equal chance of any one request being processed.
3. Static lottery bus arbiter: a probabilistic arbitration algorithm implemented in a centralized "lottery manager".
4. Dynamic lottery bus arbiter
5. LRU (least recently used) Algorithm

Bus Arbitration



- The unit requiring the bus control raises a bus-request signal to arbitration-logic, who in reply responds by bus-grant signal.
- Bus arbitration logic may be (1) a centralized control to grant the control over the bus, or may be (2) distributed. The centralized, though simple, has problem of single-point failure.

properties of Buses

- Bus is the most popular communication pathway among the various components of a computer system.
- Provides cost effective solution
- It provides set of shared communication links
- Versatile: So that new components can always be added
- It works on broadcast property
- Disadv:It is single shared communication link (no backup / standby), Bandwidth (BW) cannot increase with the increase of no. of components / units connected. May some time become communication bottleneck.
- Point-to-point communication links may be used for large BW requirements (but expensive solution).

Challenges

- Buses are pushed to provide higher data rates. This causes problems of:

  (a) Signal reflection, (b) Cross talk, (c) Skew of signal
- Cross talk? Skew Signal?(same signal reaches to different places at different times)
- Bus Physics: Electrical signal travels at finite speed (typically 5 nano sec time for one meter travel in copper wire). Clock frequency cannot be arbitrarily increased due to problem of signal reflection.
- May produce standing wave pattern due to reflection.
- Bus can be treated as transmission line. If Z_L is impedance of line, and Z_o of load, then *reflective index* is $\Gamma = (Z_L - Z_o)/(Z_L + Z_o)$. Γ is zero only if $Z_L = Z_0$, which is not possible as Z_0 is input impedance of active components. The non-zero reflective index will cause reflection of signals on the bus.

Practice Exercises

1. Determine the maximum speedup of a single-bus multiprocessor system having N processors if each processor uses the bus for a fraction f of every cycle.
2. In order to enhance the CPU-Memory interactions one solution is to have an exclusive CPU-memory bus, where communication with other sub-systems is exclusively through one of the memories meant only for that purpose.

   Alternately, one may connect Bus Adapters using which other buses which accommodate the sub-systems may be developed. Discuss the advantages and disadvantages each of these systems.
3. When Bus Adapters are used for generating more buses, we can also have one such bus exclusively for memory sub-systems, one fast bus and a slow bus for appropriate types of I/O devices. Sketch such an arrangement and discuss the modes of data transfer between the main processor-memory bus and each of these different backplane buses. Also outline how communication may be established between units across the different backplane buses.

Practice Exercises

4. Develop the asynchronous interlocked two-way communication protocol between a handshaking master and a slave involved in a write cycle, and sketch the relevant timing waveforms.
5. Assuming appropriate handshake signals, indicate the series of actions involved in priority arbitration sequence, and show how they mesh in with the on-going parallel action of data transfer.
6. A computer has 64-bit instructions, having two fields: first two bytes are for opcode, and the rest is immediate operand or operand address.
   - 6.1 What is maximum addressable memory in bytes?
   - 6.2 How many bits are required for program counter and for IR?
7. A computer has 16-bit address and 16-bit data-lines.
   - 7.1 What is maximum address space?
   - 7.2 What is size of each location in bytes?
   - 7.3 What is size of PC, AR, DR, IR?

Practice Exercises

8. Two microprocessors have 16- and 32-bit wide external data buses. Other features are same and bus cycles are identical.
   - 8.1 If all instructions and operands are 4 bytes long, by what factor the maximum data transfer rate differ?
   - 8.2 Repeat above, if half of the instructions and opcodes are two-bytes long.
9. For the synchronous read operation, the memory module must place the data on the bus sufficiently ahead of the falling edge of the Read signal to allow for the signal settling. The clock frequency is 20 MHz and Read signal begins to fall in the middle of the second half of T_3.
   - 9.1 Determine the length of the memory read cycle.
   - 9.2 When, at the latest, should memory data be placed on the bus? Allow 10 ns for settling of data lines.
10. Intel 8088 microprocessor has read bus timing like the synchronous read/write discussed in the class, but it requires 4 clock cycles. The valid data is on the bus for an amount of time that extends into the 4th cycle. Let clock is 8 Mhz.
    - 10.1 What is the maximum data transfer rate?
    - 10.2 Repeat above, assume the need to insert one wait state per byte transferred.

Practice Exercises

11. 8086 uses 16-bit bus that can transfer 2 bytes at a time, provided that lower byte has even address. However, the 8086 allows both even- and odd-aligned word operands. If odd aligned word is referenced, two memory cycles, each consisting of four bus cycles, are required to transfer the word. Consider an instruction on 8086 that involves two 16-bit operands. How long does it take to fetch the operands? Give range of possible answers. The clock is 4 Mhz and no wait state is present.