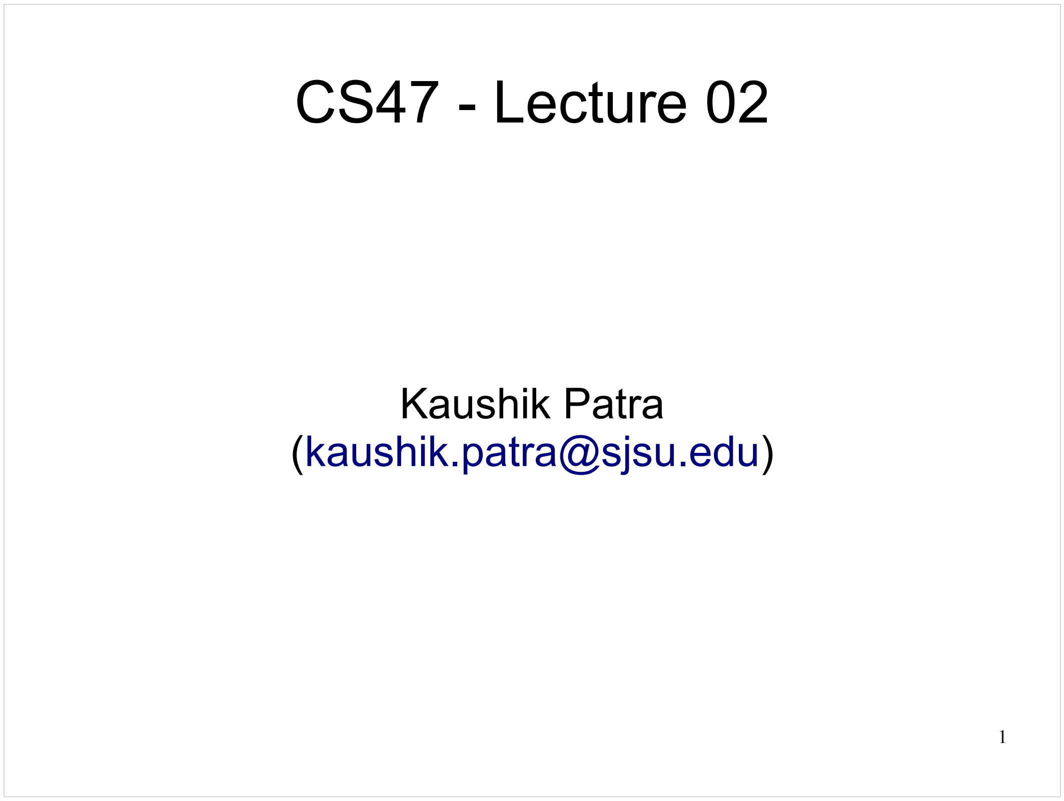
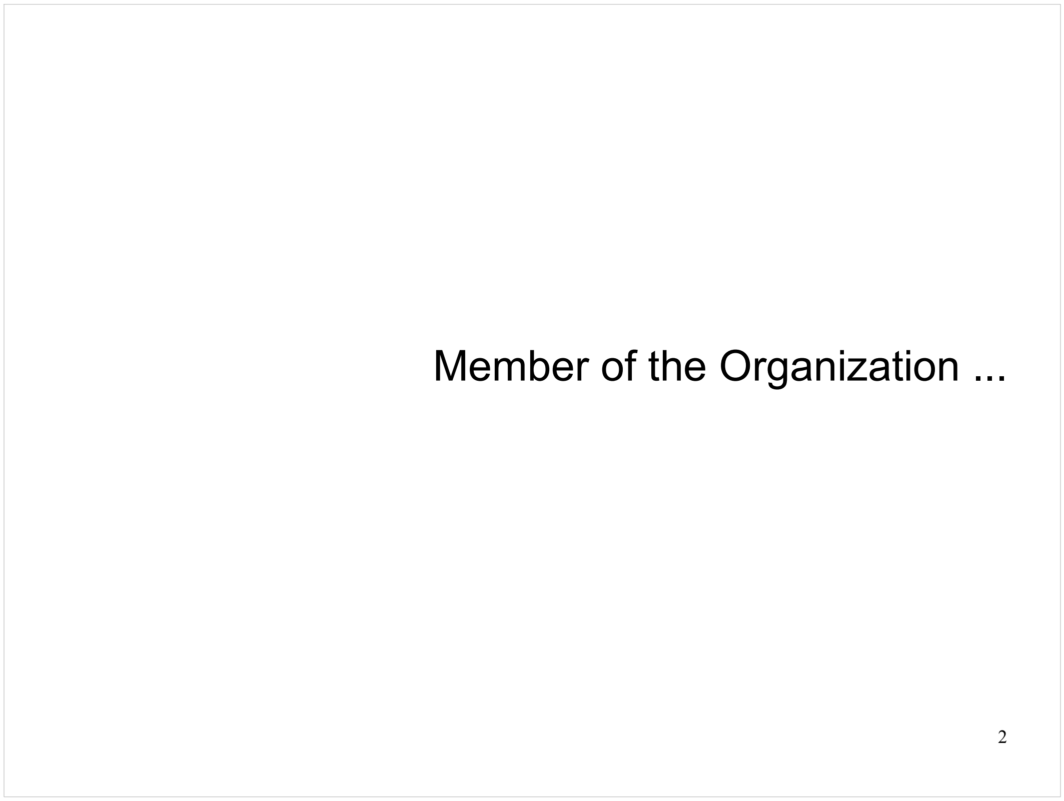


CS47 - Lecture 02

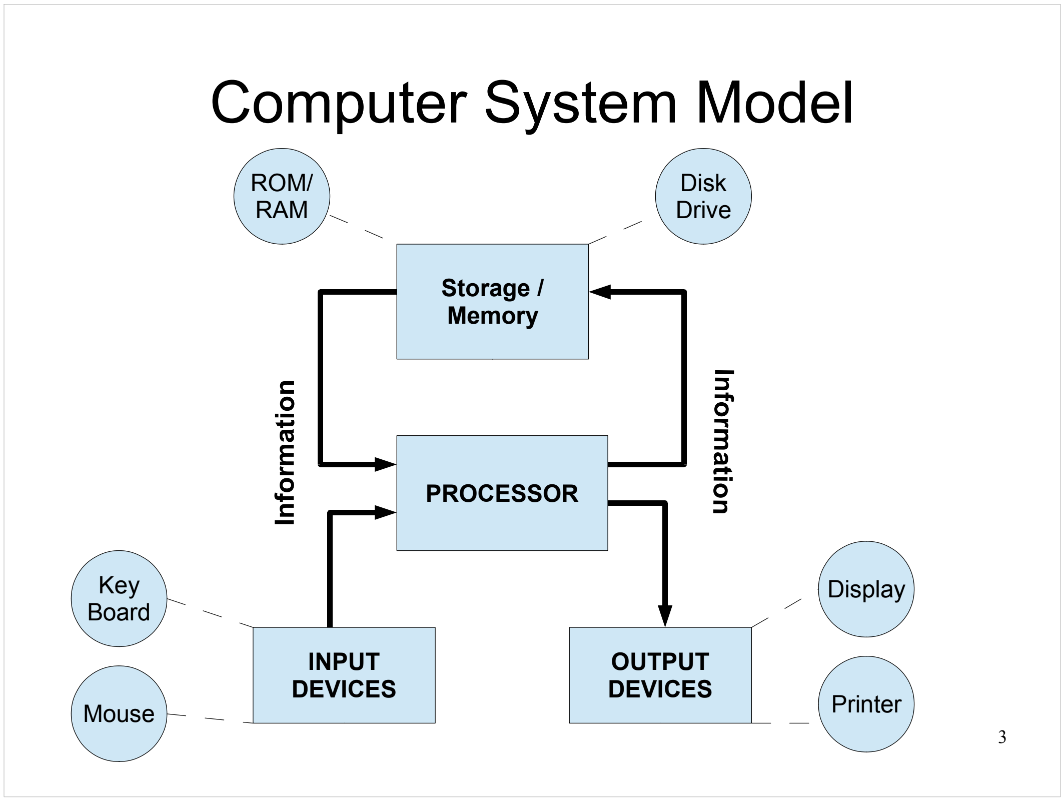
Kaushik Patra
(kaushik.patra@sjsu.edu)

1

- Topics
  - Members of the Organization
  - Clock System Model
  - Memory System Model
  - Register File Model
  - Control Unit Model



Member of the Organization ...



Computer System Model

ROM/ RAM

Disk Drive

Storage / Memory

Information

PROCESSOR

Information

Key Board

Mouse

INPUT DEVICES

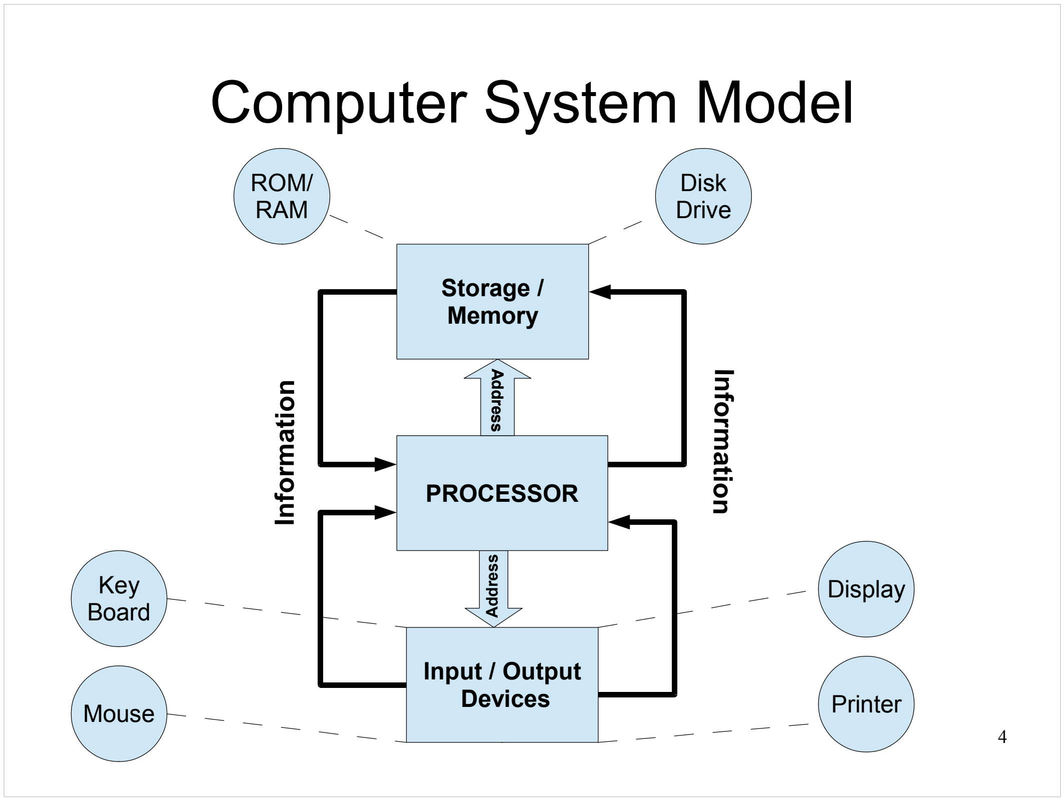
OUTPUT DEVICES

Display

Printer

3

- A typical computer system has following major components.
  - Processor : Do the main computing, execute program written into memory.
  - Storage / Memory: Stores information, can be fetched as needed. Can be volatile (RAM) or non-volatile (ROM / Disk Drive) – Can be semiconductor based (RAM/ROM) or mechanical (Disk drive).
  - Input Devices: To send in information to computer (Keyboard, mouse, joystick, ...)
  - Output Devices: To retrieve information from computer (display, printer, ...)
- Information are either instructions or data to be processed by processor. Information flows into processor from input device and memory. Information flows out from processor to output devices or memory.



Computer System Model

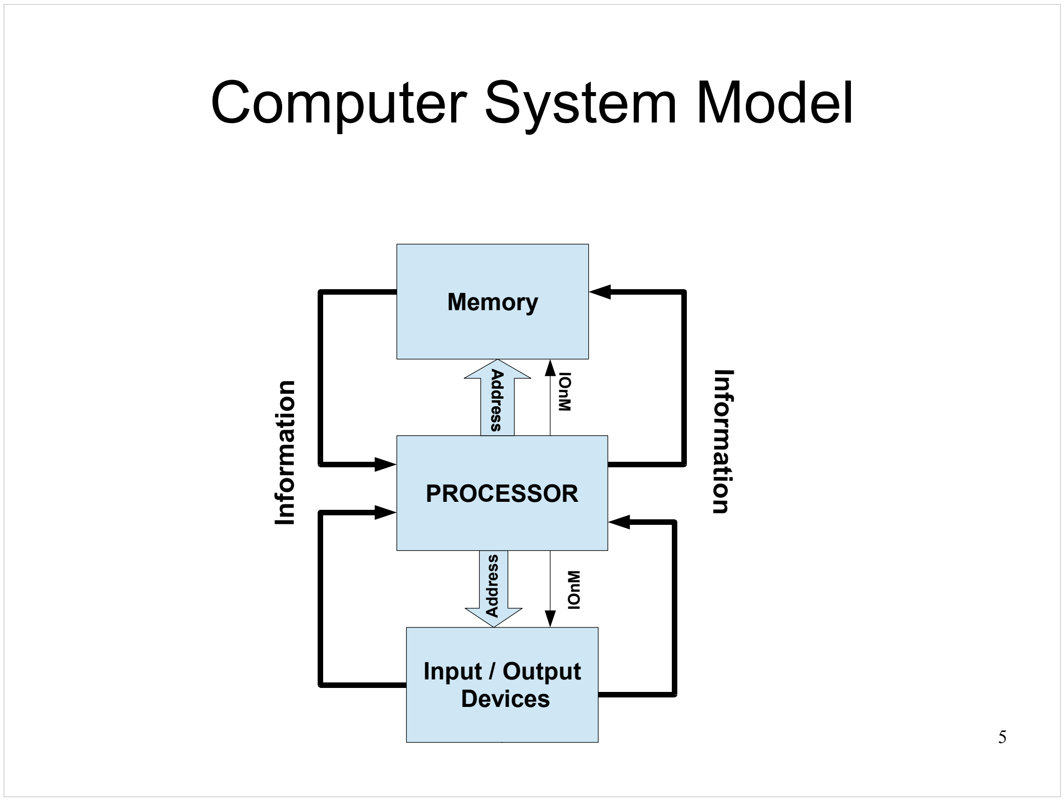
- If we combine groups of input and output devices, conceptually we have IO devices in a group. If we do so, information flow is similar between IO devices and memory/storage devices (in fact, disk drive can be thought of another IO device type). Information flows in and out from IO devices or storage / memory devices.

- Processor is the active device who wants to fetch data (the term 'data' and 'information' is interchangeably used in the course reader without any change in underlying concept) from memory / storage or IO devices. Typically data is not pushed from devices into processor – processor fetches data as needed.

- Processor issues unique number to locate information within storage / memory or IO devices. This unique number is called 'addresses' which has similar concept of postal address used in real life.

- Usually a modern day a computer with 32 bit can generate 2^{32} unique addresses which is equivalent to having 4G (4294967296) unique addresses. The set of addresses is called address space of the computer. In this case of 32-bit addressing has address space of 0 to 4294967295.

  - A computer with 64 bit has address space of virtually infinity size. Now a days processor generates 40 bit worth addresses which can address 1TB address space.



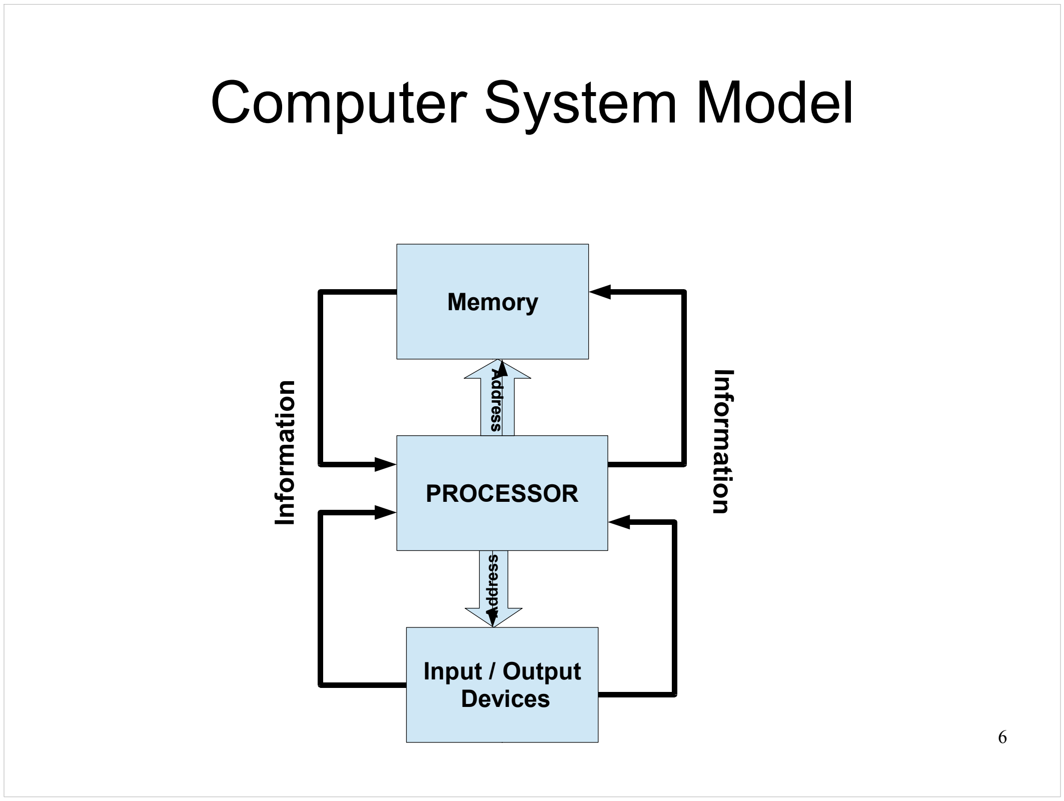
Computer System Model

- We can model storage (like disk drive) similar to memory from processor perspective. There are certain differences in access mechanism, but for our discussion this simplification works well – we'll treat them same.

- To distinguish address issued for IO devices vs. issues for memory, a special signal IOnM is issued from processor. The 'IOnM' is used for our discussion purpose, in real life it can assume a different name. Usually this name suggests that if the signal value is logic 1, the address issued is for IO devices and if it is logic 0 then the address issued is for memory devices.

  - Processor can issue same address to access information from memory or IO devices depending on IOnM value.

- This method of addressing IO devices using a special signal is known as '**IO mapped IO**'. In older days it was a significant technique since address space of a computer was very limited (8 or 16 bits). By distinguishing address of IO device from address of memory, it enabled older generation processor to access larger address space.

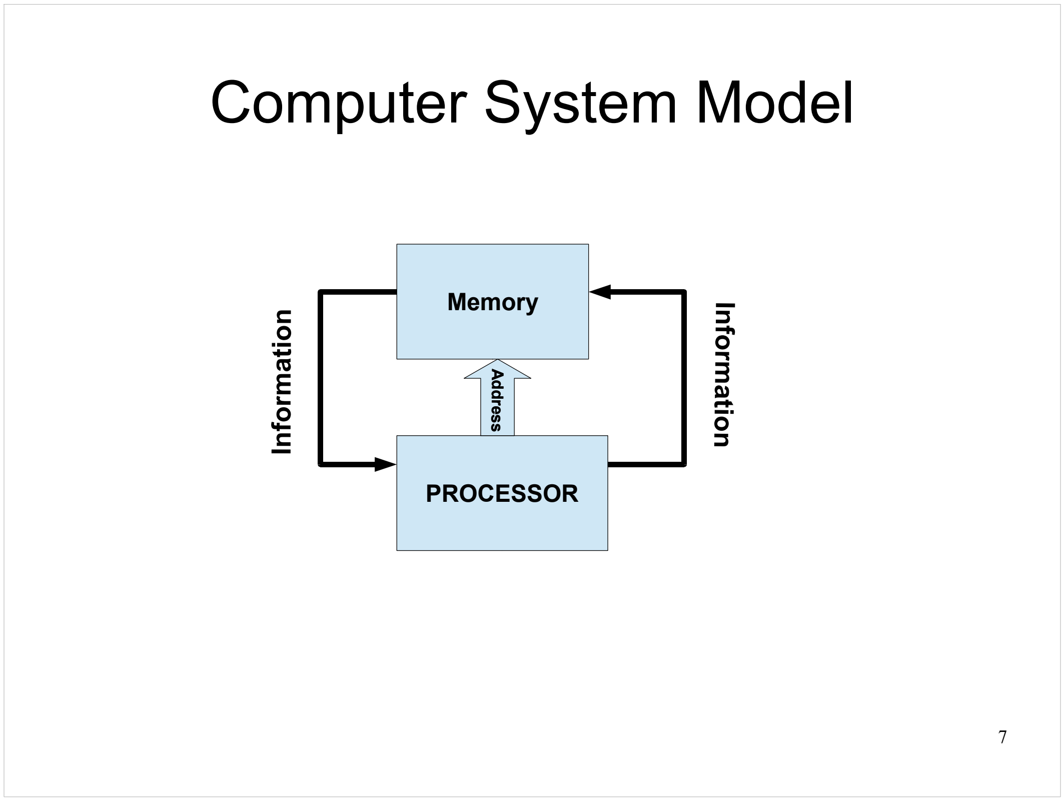


Computer System Model

- Modern days computer has huge address space (32-bit or 64-bit) that is more that sufficient. In this scenario, instead of having yet another control signal from processor, conceptually the control signal can be made a part of the address itself. In that way, we can save extra complexity due to having a separate control signal (even reducing one signal make significant reduction is circuit complexity in real hardware implementation).

- In this technique, an address can either be for a IO devices, or for a memory not for both, unlike the 'IO mapped IO.' IO device addresses are part of entire address space that can be addressed by the processor. Information flow from IO devices is treated same as information flow from memory devices. This technique is called 'Memory mapped IO.'

  - From the processor perspective, it does not matter if processor wants to fetch a data from memory or a keyboard (for example) – it issues an address and wait for a data to comeback from that address. It is hardware design which makes distinction if the data should come from RAM/ROM or keyboard.



Computer System Model

Memory

Address

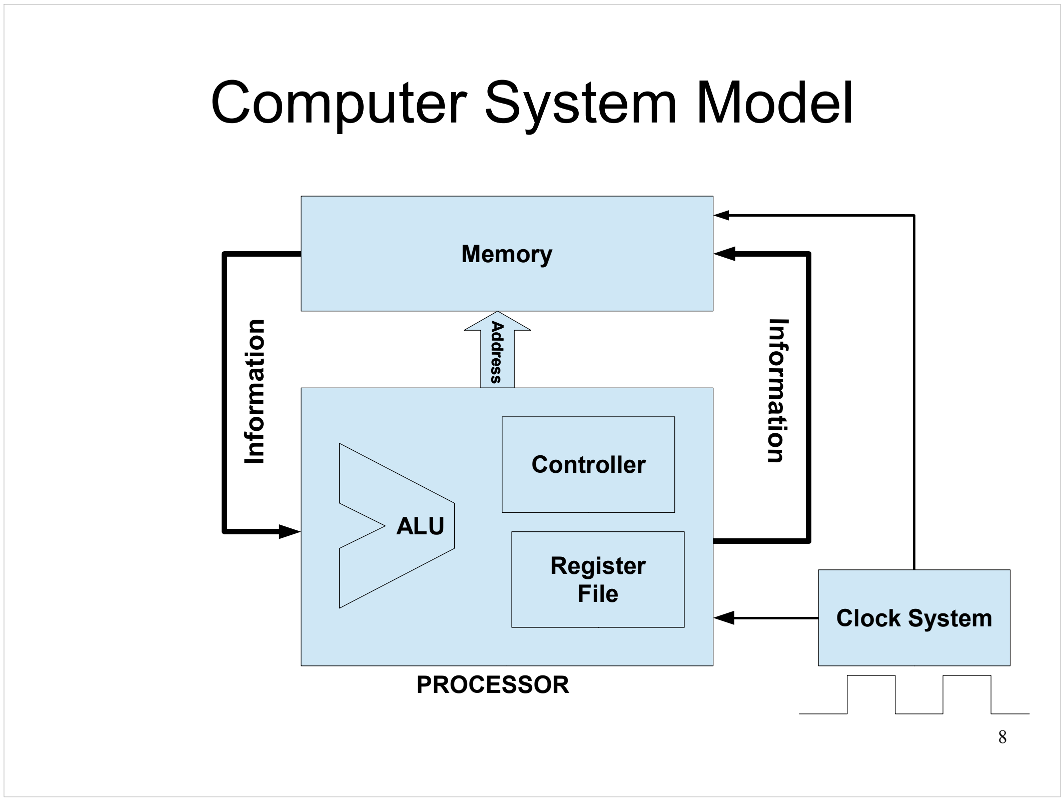
PROCESSOR

Information

Information

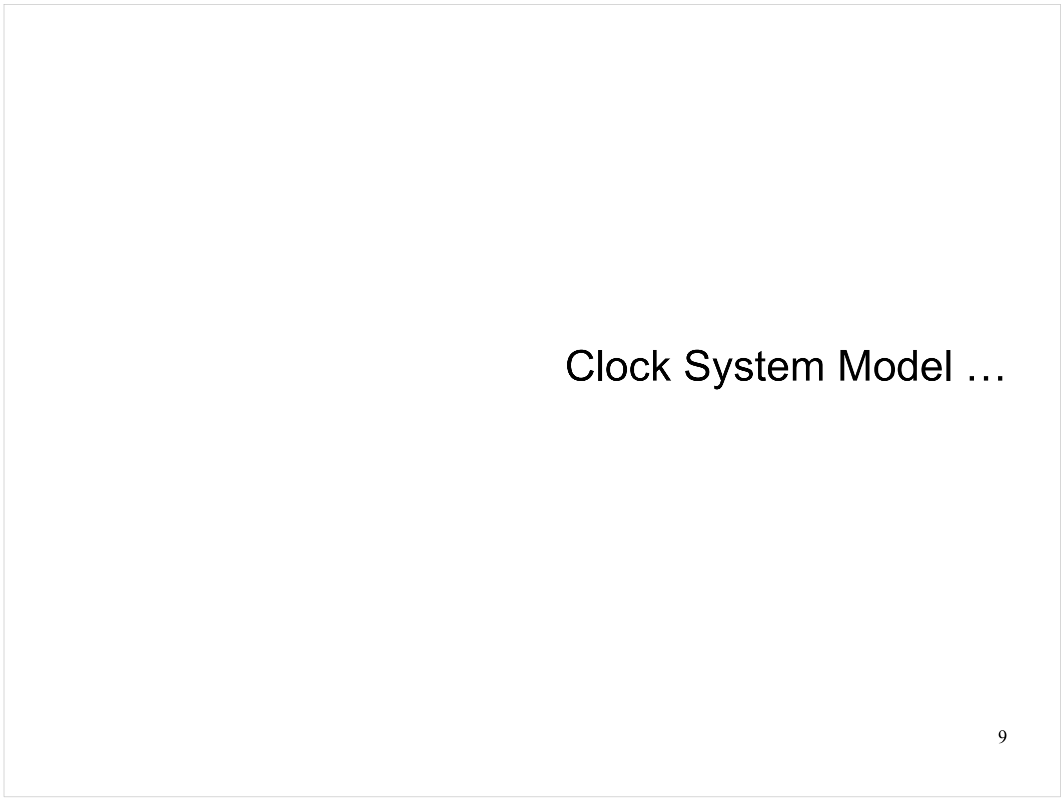
7

- Most of the modern days computer uses memory mapped IO. Therefore, we can treat IO devices and memory same during our discussion. This will simplify the model that we need to understand.

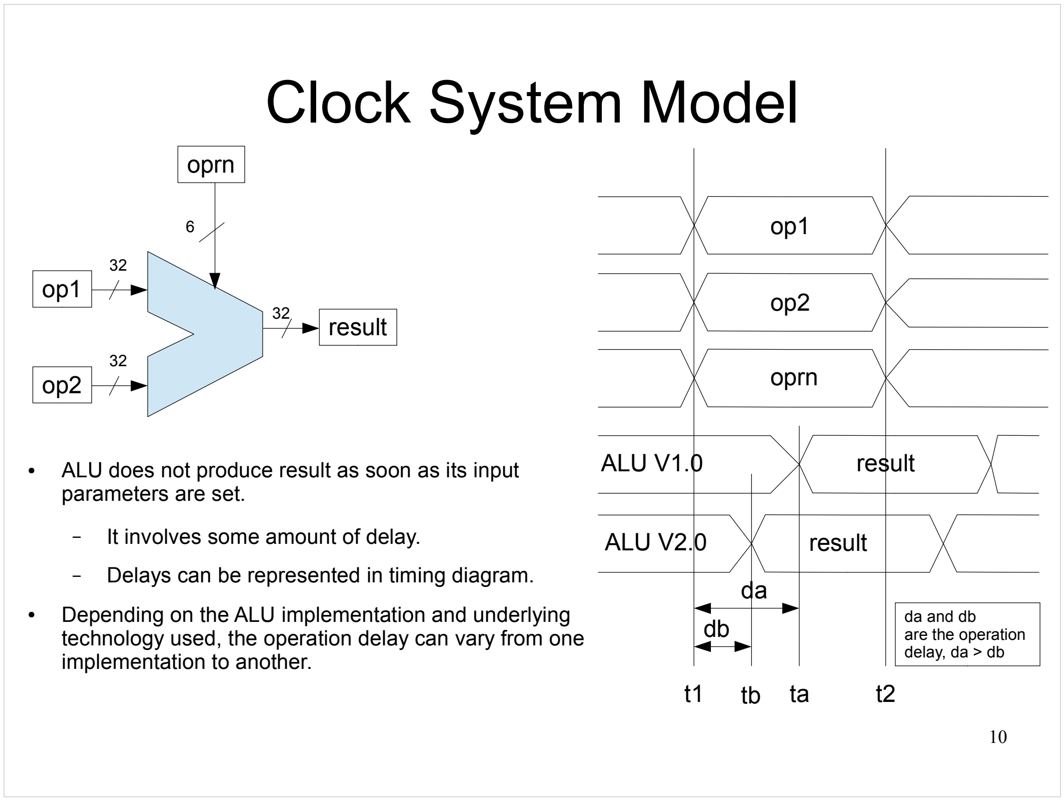


Computer System Model

- Now if we dive down into a processor we can find three major components.
  - **ALU**: Arithmetic and Logic Unit which performs all basic arithmetic and logic operations that a processor supports.
  - **Controller**: This is the sub-system which synchronizes the operation between different subsystem of the processor and memory.
  - **Register File**: This is small amount of very fast memory (usually 32 or 64 in total space) inside the processor to be used as temporary storage for basic operations.
- The clock subsystem provides continuous stream of logic 1 and 0 at specific interval. This signal is used to synchronize operation between different subsystem of the computer. All the modern computer uses clocks. The hardware design style using clock signal as reference to synchronize operations is called 'synchronous design'.



Clock System Model …



Clock System Model

- ALU does not produce result as soon as its input parameters are set.
  - It involves some amount of delay.
  - Delays can be represented in timing diagram.
- Depending on the ALU implementation and underlying technology used, the operation delay can vary from one implementation to another.

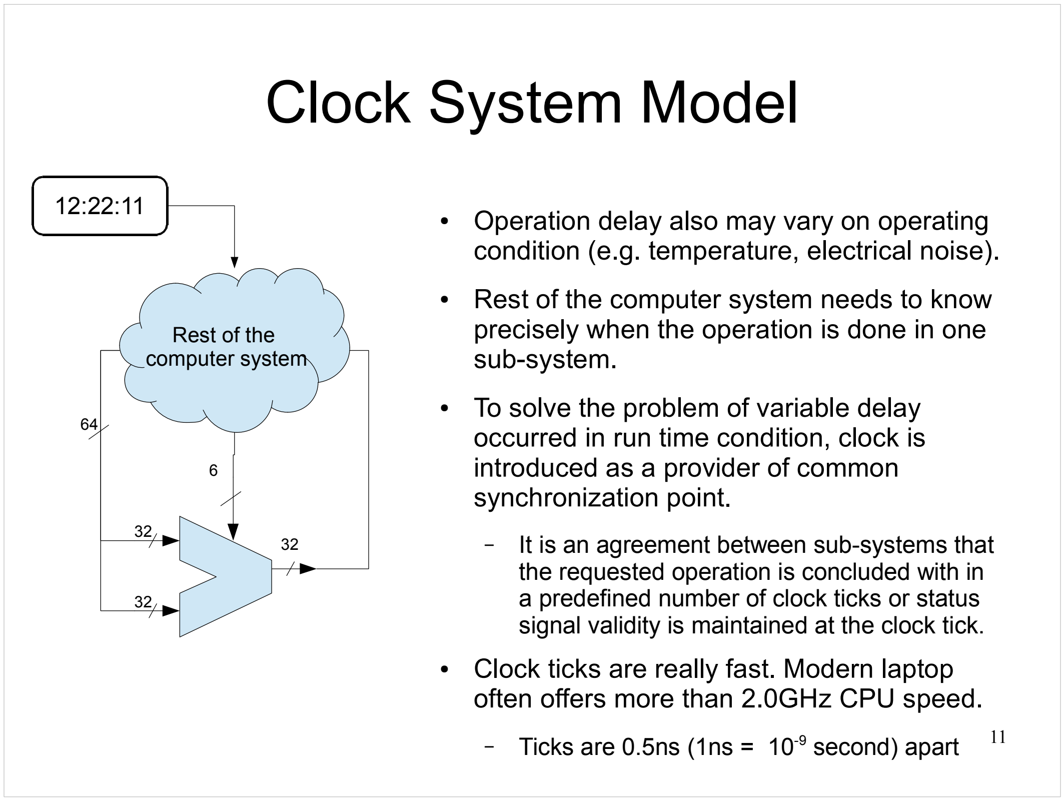
da and db are the operation delay, da > db

- Speed of operation for any digital logic circuit is limited by the underlying devices and technology that implements it. It is a part of electrical engineering study to learn techniques to speed up logic operations.

- Speed of operation also depends on the implementation of the logic circuit. Like a software implementation, one set of hardware objective (e.g. providing ALU functionality) can be done through different implementation. It is part of computer science study to improve algorithms for faster operation.

- Timing diagram depicts change of values in parameters or signals in association with time. Simplistic model assumes that the value change is instantaneous. However, in reality there are delay associated even with value change.

- Group of multiple signal (or bus) is depicted with hexagon in timing diagram, where single signal is depicted with single line.


12:22:11
Rest of the computer system
64
6
32
32
32

Clock System Model

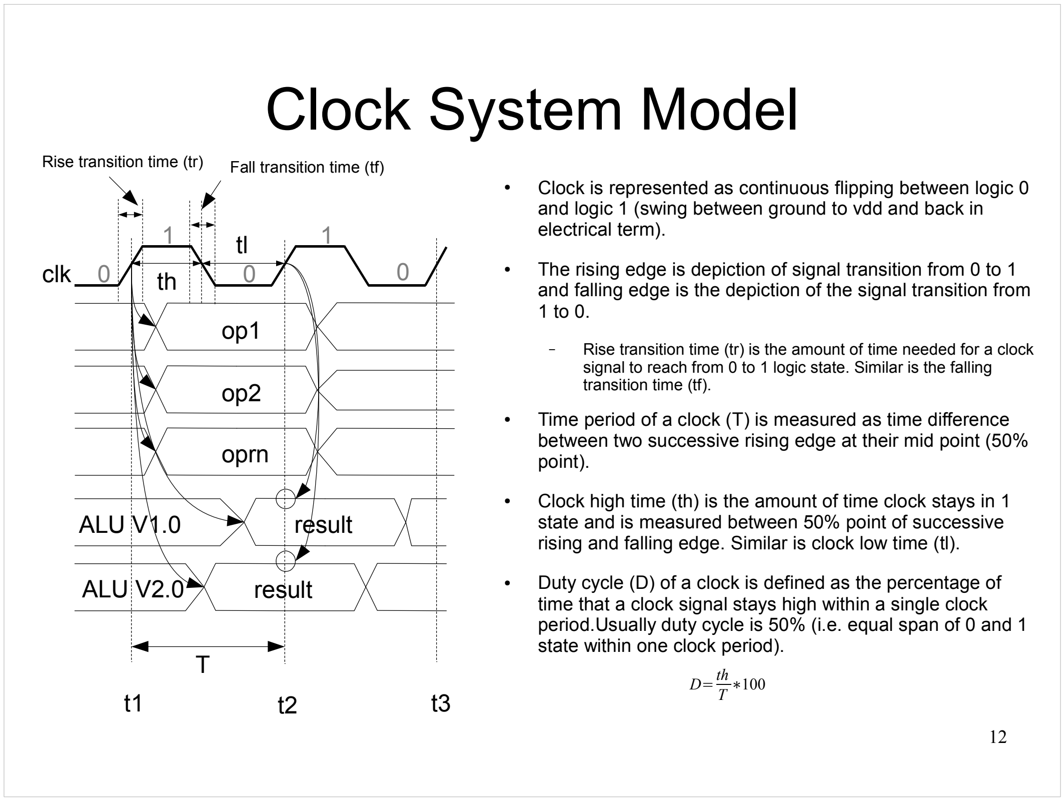
- Operation delay also may vary on operating condition (e.g. temperature, electrical noise).
- Rest of the computer system needs to know precisely when the operation is done in one sub-system.
- To solve the problem of variable delay occurred in run time condition, clock is introduced as a provider of common synchronization point.
  - It is an agreement between sub-systems that the requested operation is concluded with in a predefined number of clock ticks or status signal validity is maintained at the clock tick.
- Clock ticks are really fast. Modern laptop often offers more than 2.0GHz CPU speed.
  - Ticks are 0.5ns (1ns = 10^{-9} second) apart

- All the subsystems of a computer synchronizes their operation at the clock tick. In digital logic term, this type of circuits involving synchronization by clock is known as 'synchronous' circuit.

- There is another class of digital design not involving common synchronization point, known as 'asynchronous circuit'.

- Though asynchronous circuit can, theoretically, achieve much higher speed of operation, it is very hard to implement complex yet highly reliable circuit with that technique. On the other hand, though we are sacrificing operation performance, it is easier to implement synchronous circuit with predictable behavior.

- Distance between the clock ticks (or time period) of clock is defined as reciprocal of clock frequency (**T = 1/f**).



Clock System Model

- Clock is represented as continuous flipping between logic 0 and logic 1 (swing between ground to vdd and back in electrical term).
- The rising edge is depiction of signal transition from 0 to 1 and falling edge is the depiction of the signal transition from 1 to 0.
  - Rise transition time (tr) is the amount of time needed for a clock signal to reach from 0 to 1 logic state. Similar is the falling transition time (tf).
- Time period of a clock (T) is measured as time difference between two successive rising edge at their mid point (50% point).
- Clock high time (th) is the amount of time clock stays in 1 state and is measured between 50% point of successive rising and falling edge. Similar is clock low time (tl).
- Duty cycle (D) of a clock is defined as the percentage of time that a clock signal stays high within a single clock period.Usually duty cycle is 50% (i.e. equal span of 0 and 1 state within one clock period).

$$D=\frac{th}{T}*100$$

12

- Clock in real circuit is made using quartz crystal. Its mechanical resonance of vibration is translated into electrical signal oscillation through electronic oscillation circuit. This mechanism can produce precise amount of clock frequency needed for the target system.

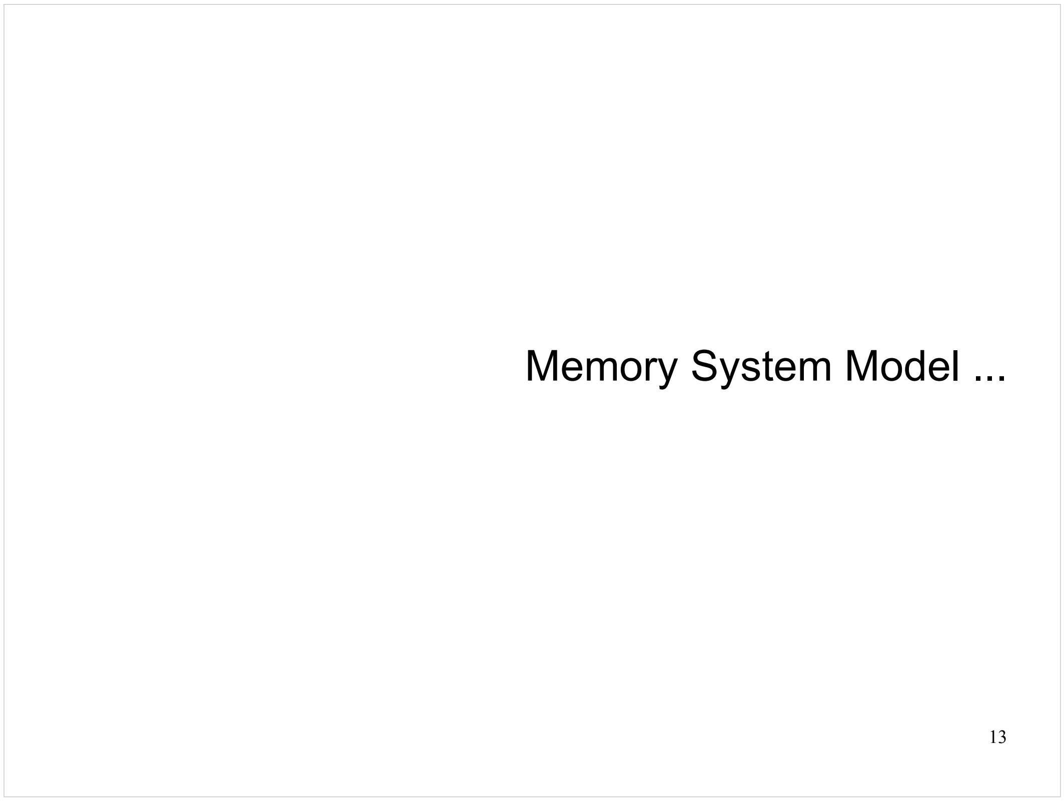
- Time period T is sum of clock high time and low time (T = th + tl).

- In the timing diagram it is shown using connecting arrows that rising clock edge at t1 triggered the change in op1, op2, oprn, and result. At the rising edge at t2 the result is sampled / used.

  - We usually use this type of relational arrow for multi clock cycle operations to clearly denote which clock edge triggered what event.

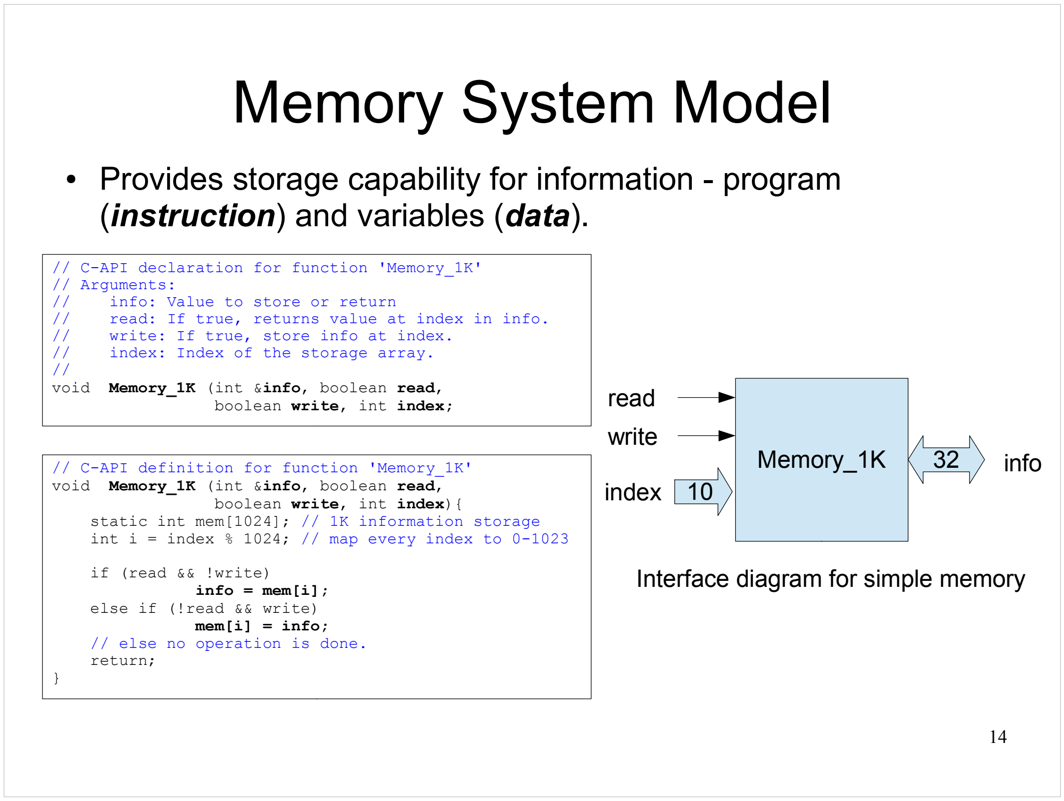
- Operation can be synchronized at either rising or falling edge.

- Since result is sampled at certain predefined interval, synchronous circuit sacrifices certain amount of performance (since it is not using the result as soon as it is ready). However, synchronous design can give much more stability to the digital circuit.



Memory System Model ...

13



Memory System Model

- Provides storage capability for information - program (***instruction***) and variables (***data***).

```
// C-API declaration for function 'Memory_1K'
// Arguments:
//    info: Value to store or return
//    read: If true, returns value at index in info.
//    write: If true, store info at index.
//    index: Index of the storage array.
//
void  Memory_1K (int &info, boolean read,
                 boolean write, int index;
```

```
// C-API definition for function 'Memory_1K'
void  Memory_1K (int &info, boolean read,
                 boolean write, int index){
    static int mem[1024]; // 1K information storage
    int i = index % 1024; // map every index to 0-1023

    if (read && !write)
               info = mem[i];
    else if (!read && write)
               mem[i] = info;
    // else no operation is done.
    return;
}
```

Interface diagram for simple memory

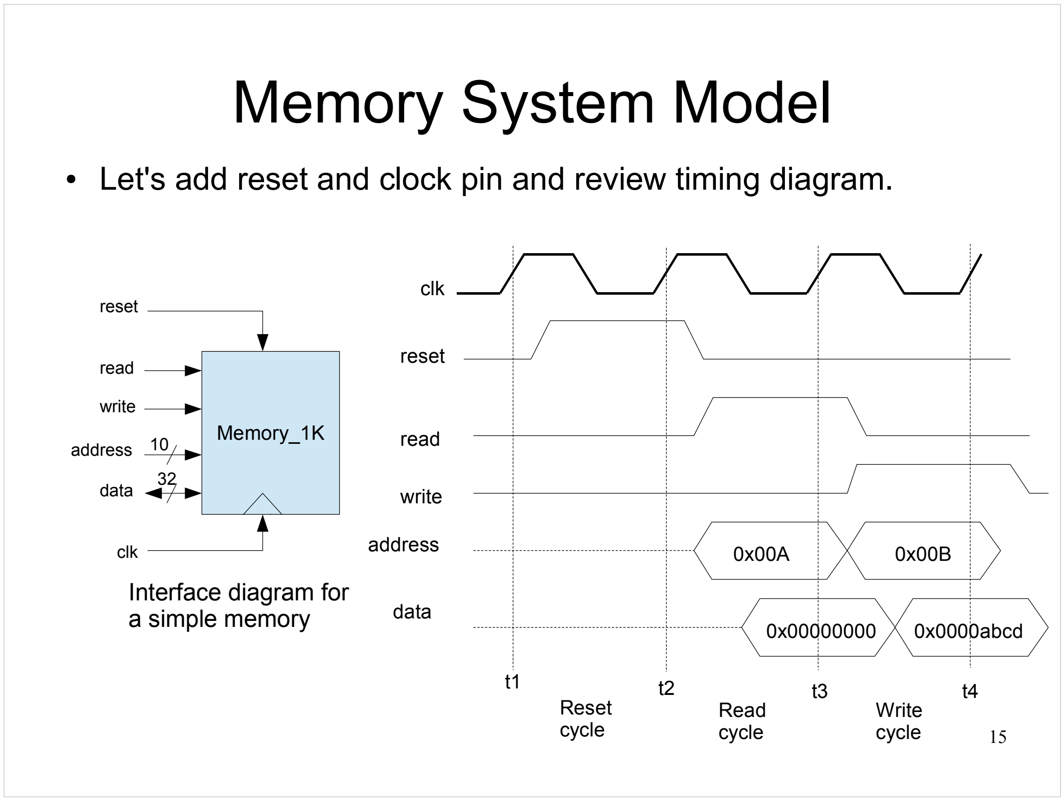
- static variable within a function defines a non-volatile storage within scope of that function.

- There is no distinction between 'instruction' and 'data' in context of storage. Any one of them is 'data' from memory perspective. They are just a sequence of bit values. We usually call 'info' as 'data' in general in memory context.

- In context of memory, the 'index' is usually called 'address'.

- There is usually additional 'reset' pin to clear memory content and set all index position to 0.

- This is very simplistic model of memory where transaction happens as 32-bit long word. In most cases, memories are byte addressable (i.e. specific byte in an word can be read/write).



Memory System Model

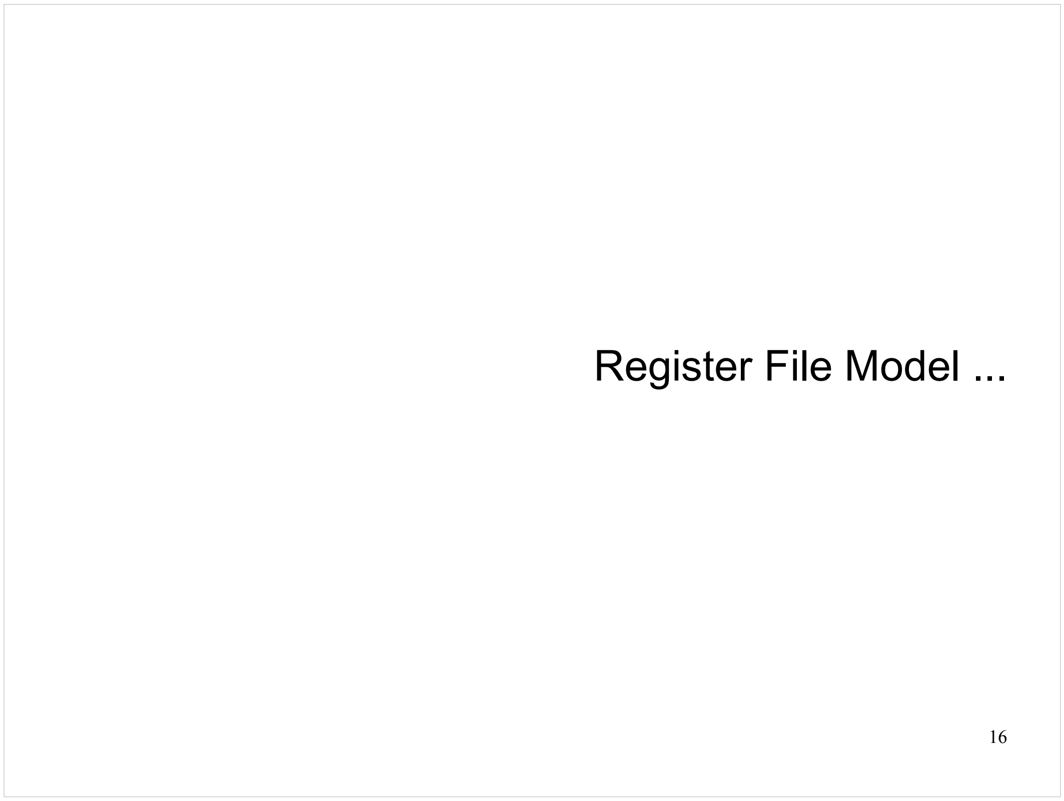
- Let's add reset and clock pin and review timing diagram.

- When both read and write are of same value, the data bus is on High-Z state which means it is electrically detached (open circuit) from rest of the circuit.

- The timing diagram shows that at t1 time reset operation is done following by a read operation at t2 and write operation at t3.

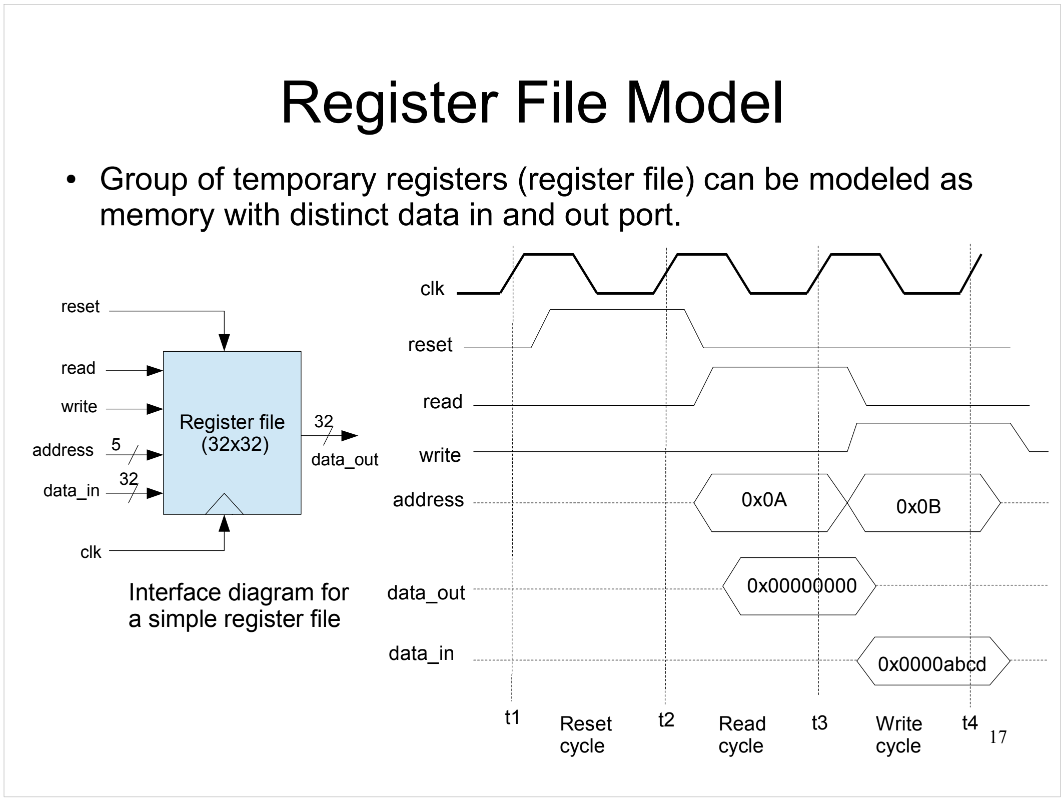
- This memory provides the instruction and data storage for a program to run on a processor.

- Memory may need multiple clock cycle to complete its operation. We have modeled a single clock cycle access to simplify system implementation.



Register File Model ...

16



Register File Model

- Group of temporary registers (register file) can be modeled as memory with distinct data in and out port.

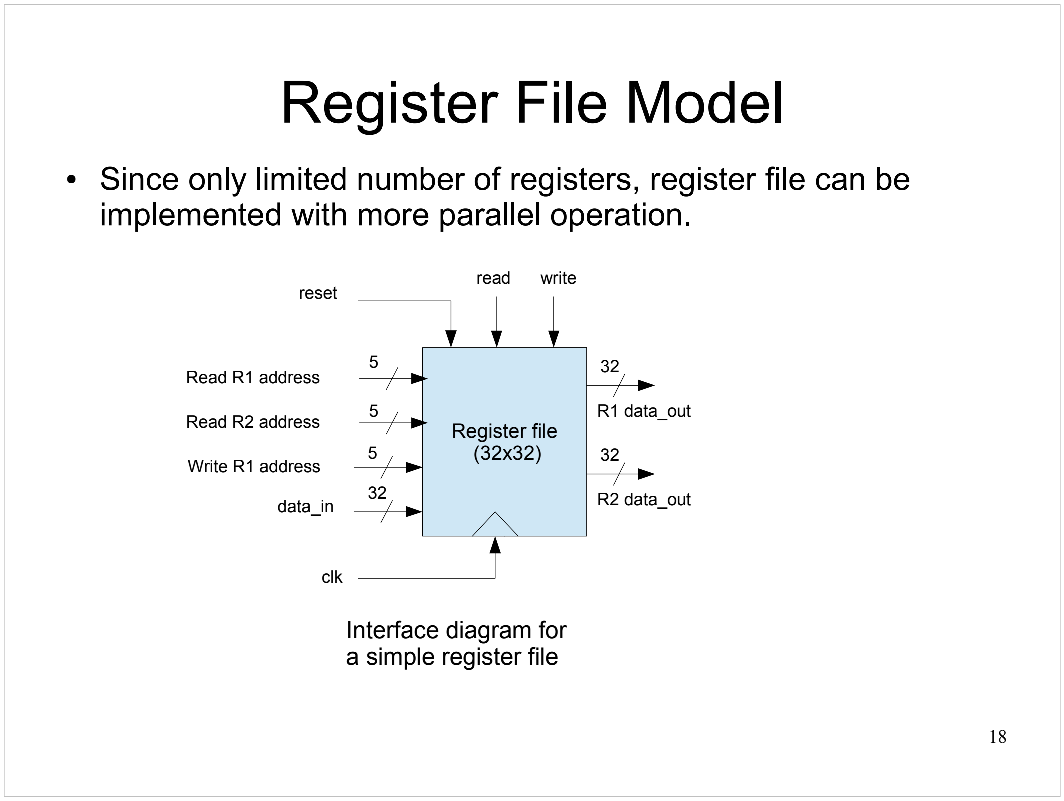
Interface diagram for a simple register file

17

- When not reading, the data_out remains High-Z.

- Register file is given separate data in and data out to simplify implementation of processor, unlike the standard memory model. Standard memory model targets to save some space and wiring using common in-out data bus for data operation.

- Register file is usually a part of processor, where standard memory is modeled outside processor.
  - We'll review concept of 'cache' memory which sits inside processor as an integrated part of it.



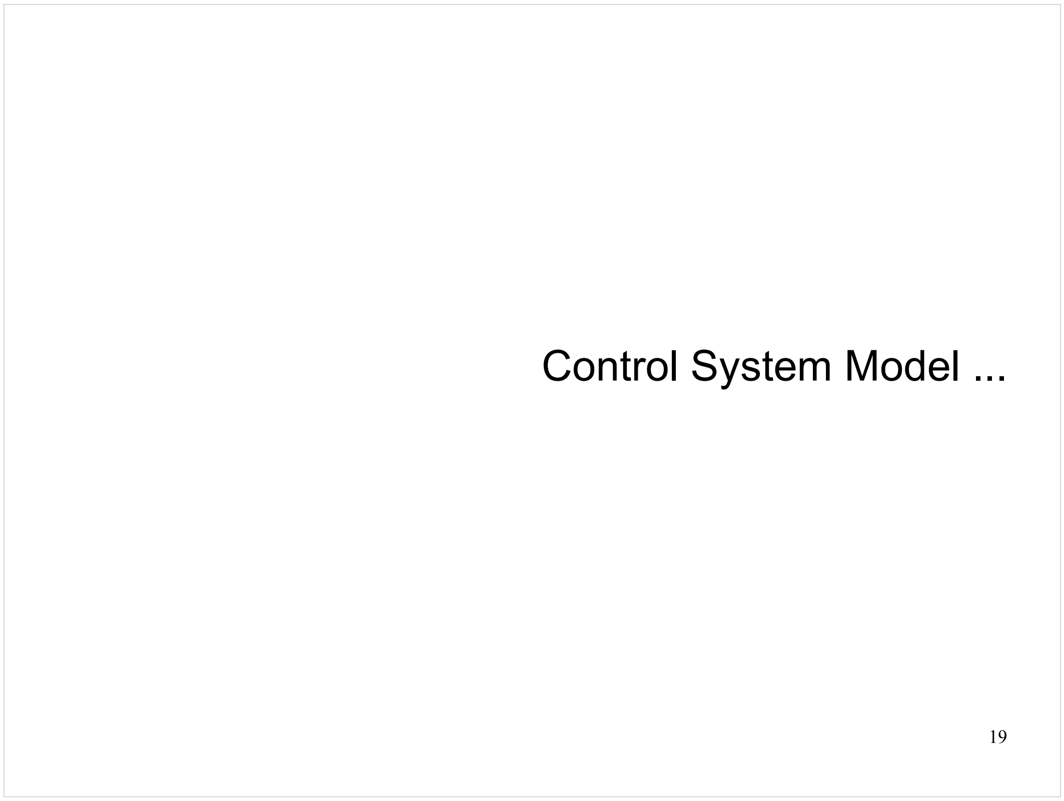
Register File Model

- Since only limited number of registers, register file can be implemented with more parallel operation.

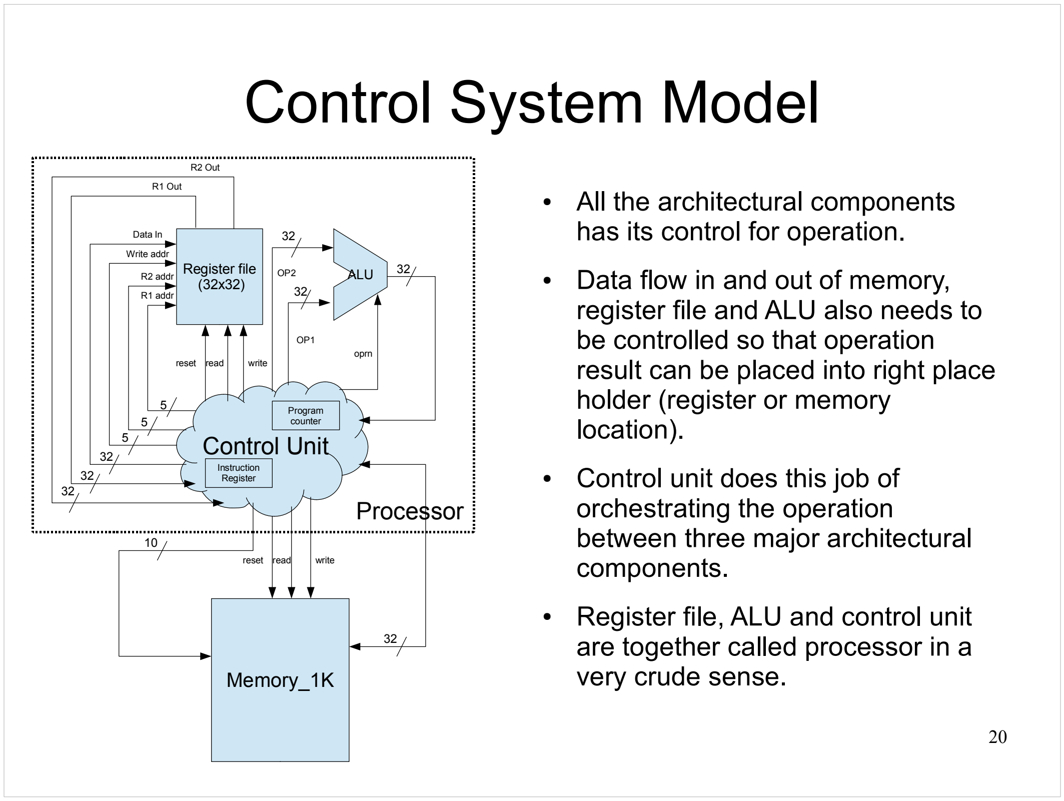
Interface diagram for a simple register file

18

- Since our instruction set 'cs147sec05' has fixed position for the register operands, it will be easier and faster to access both the register at once.
  - This is instruction set driven architecture.
  - If the instruction set does not provide such uniformity, there would no such simplification for faster operation.



Control System Model ...



Control System Model

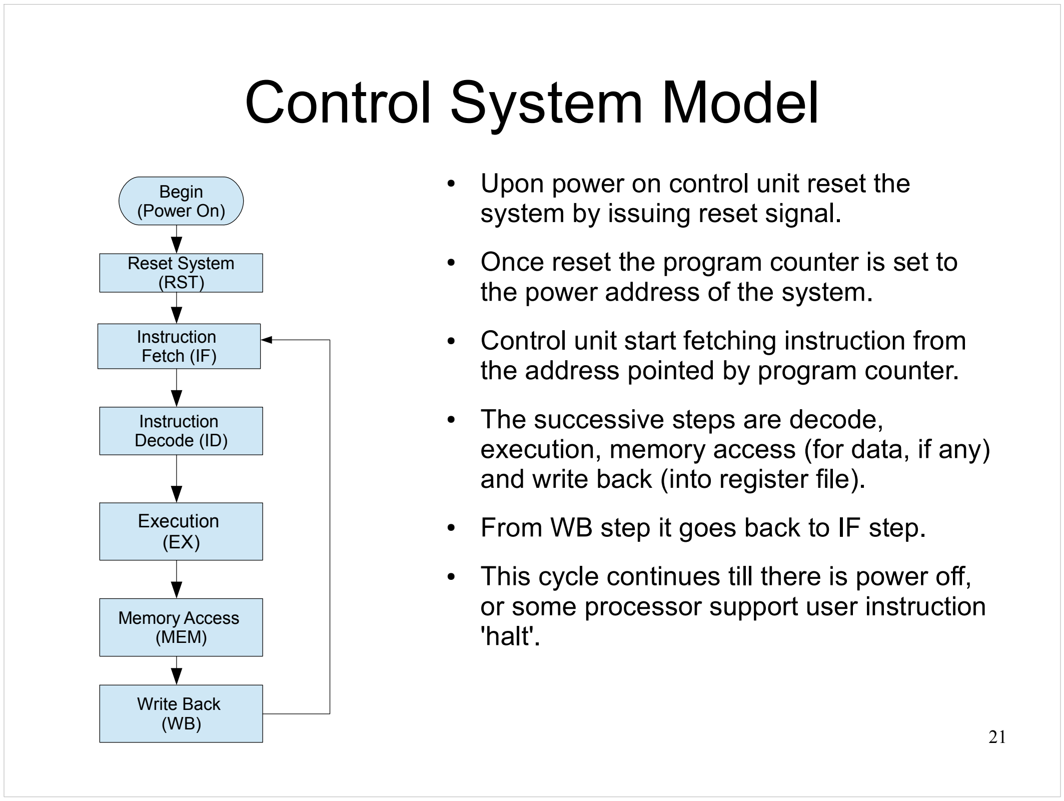
- All the architectural components has its control for operation.
- Data flow in and out of memory, register file and ALU also needs to be controlled so that operation result can be placed into right place holder (register or memory location).
- Control unit does this job of orchestrating the operation between three major architectural components.
- Register file, ALU and control unit are together called processor in a very crude sense.

20

- Both register file and memory has reset, read and write signal. ALU has 'oprn' signal to select the operation.

- Data flow occurs between
  - Register file to ALU as op1 and op2
  - ALU to register file as result.
  - Register file to memory to store result.
  - Memory to register file to load value from memory.

- Control unit contains two special registers.
  - Instruction register to hold current instruction after fetching from memory.
  - Program counter to hold memory address to fetch next instruction.



Control System Model

Begin (Power On)

Reset System (RST)

Instruction Fetch (IF)

Instruction Decode (ID)

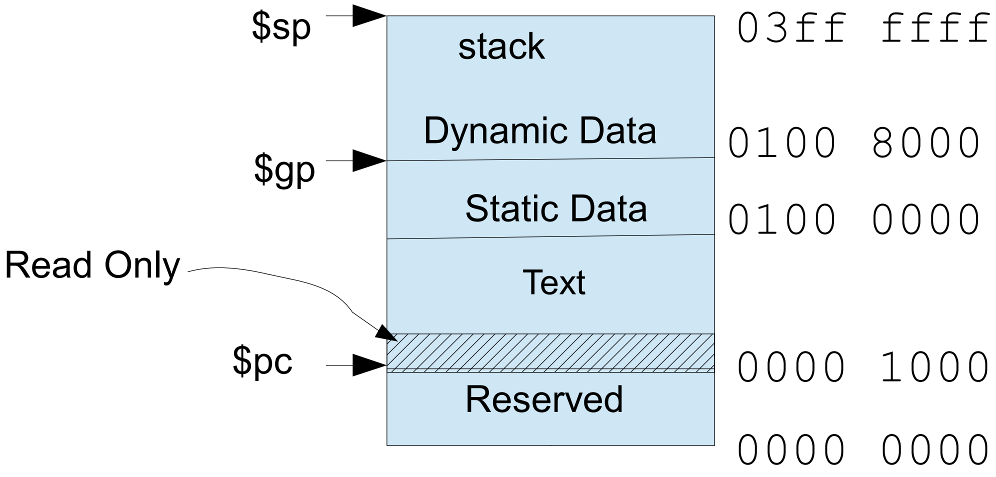
Execution (EX)

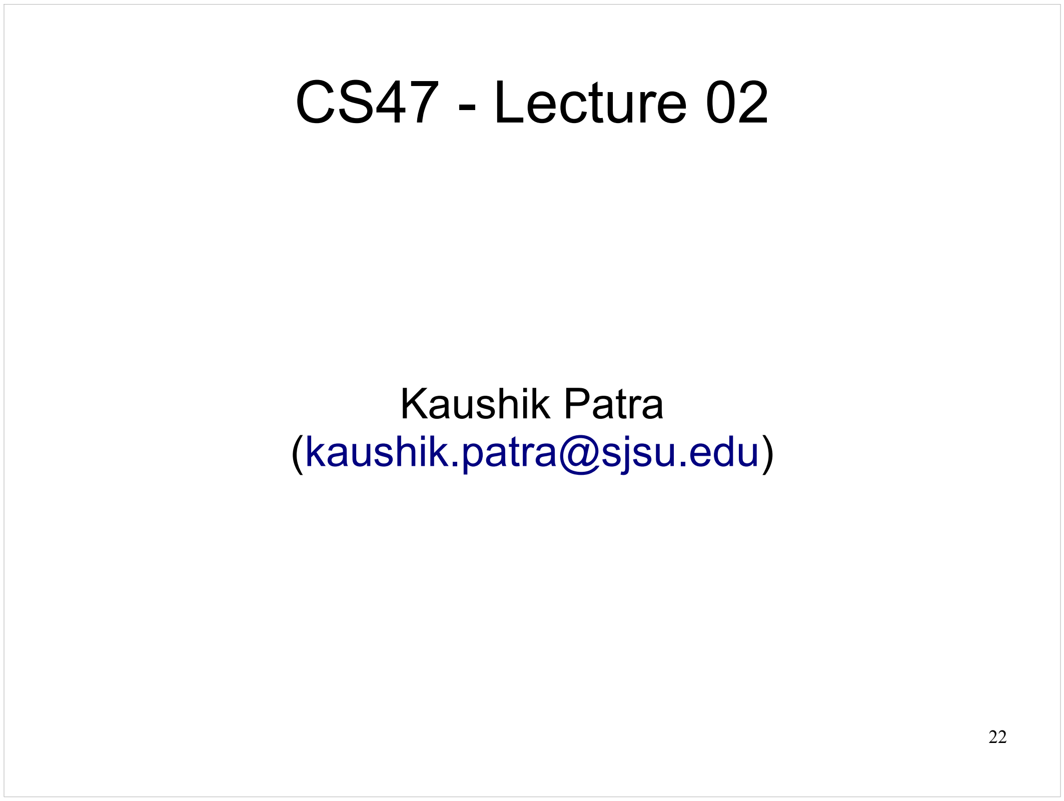
Memory Access (MEM)

Write Back (WB)

- Upon power on control unit reset the system by issuing reset signal.
- Once reset the program counter is set to the power address of the system.
- Control unit start fetching instruction from the address pointed by program counter.
- The successive steps are decode, execution, memory access (for data, if any) and write back (into register file).
- From WB step it goes back to IF step.
- This cycle continues till there is power off, or some processor support user instruction 'halt'.

- Power on address is predefined for a system.
- A part of memory starting from the power address are implemented as read-only memory (ROM) which can not be rest with a reset signal. This read-only memory part contains initial set of instructions to load BOOT program (from disk or other semi permanent memory) at the start of volatile part of the system memory (RAM) within the text area of the system memory map.





CS47 - Lecture 02

Kaushik Patra
(kaushik.patra@sjsu.edu)

- Topics
  - Members of the Organization
  - Clock System Model
  - Memory System Model
  - Register File Model
  - Control Unit Model