

ME 311: Microprocessors and Automatic Control

**Introduction to microprocessors:
Putting things together**

Acknowledgements: Anand (Mtech 09)+



**P.S. Gandhi
Mechanical Engineering
IIT Bombay**

PRASANNA S GANDHI
gandhi@me.iitb.ac.in

1



Five aspects of Education

- Information,
- Concepts,
- Innovation,
- Attitude,
- Freedom

PRASANNA S GANDHI gandhi@me.iitb.ac.in

2



Today's class

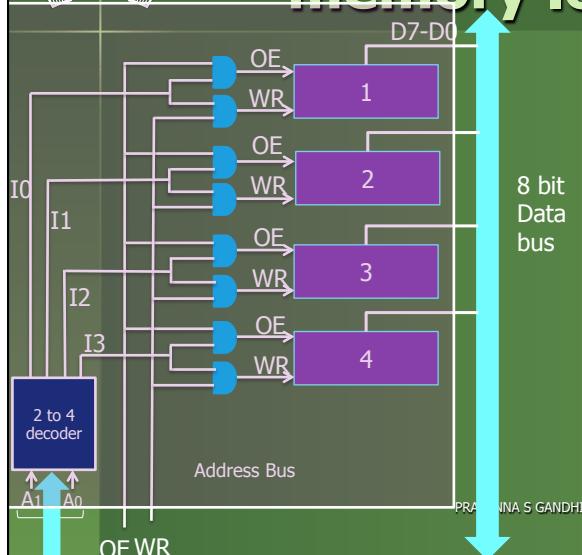
- Basic building blocks: combinational circuits and sequential circuits ← already discussed
- Memory using D FFs ← we saw in last class
- How to put things together? A primitive microprocessor
- Adding on more sophisticated timing and control

PRASANNA S GANDHI gandhi@me.iitb.ac.in

3



Recap: Addressing memory location



- How decoder can be constructed?
- Combinational logic! Think

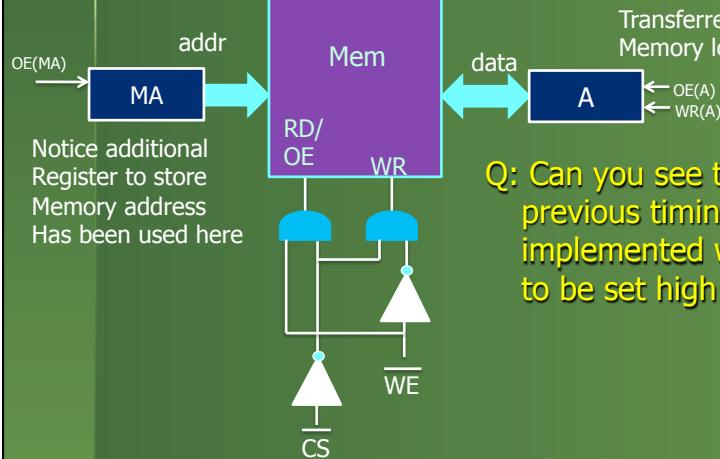
PRASANNA S GANDHI



Recall timing diagram and 'write operation' from last class

Physical connections

This is external register
From which data is to be
Transferred to specified
Memory location



Q: Can you see to have previous timing diagram implemented what pins are to be set high or low?

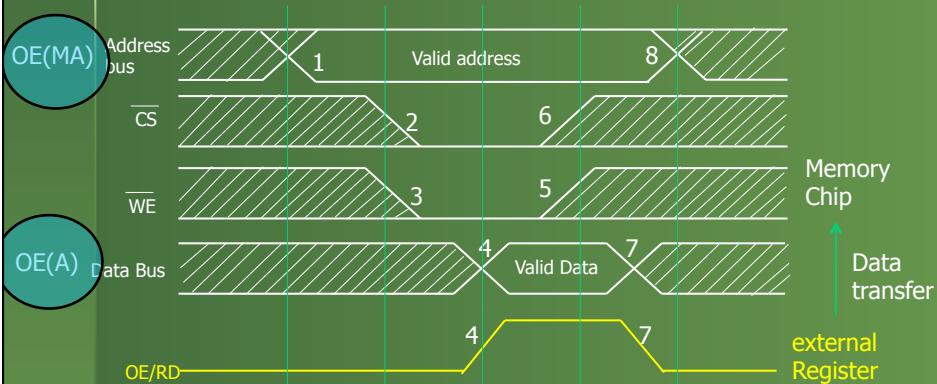
PRASANNA S GANDHI gandhi@me.iitb.ac.in

5



Timing diagram for write operation

to memory at location given by valid address from external register.



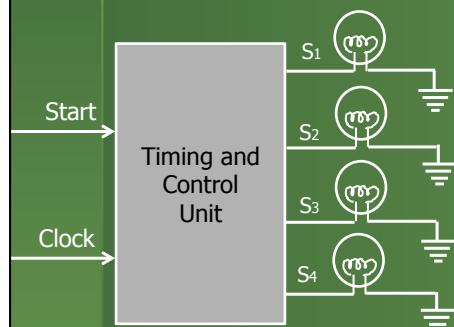
Q: How do we automate this process? Can you think of a way?

Hint : Counters

Hmmn!! How do we use them for this sequence to be executed automatically!⁶



Generation of sequence of pulses



- We want the bulbs to light one after the other
- One way: ring counter with reset as discussed → asynchronous
- Other way: D-FF with clear input (when this input is high then no matter what previous output is the current output is set to 0)
- Can you think of circuit??

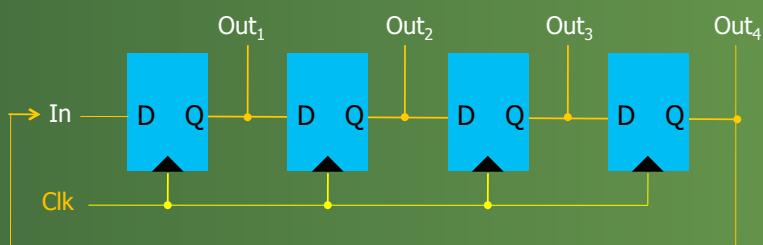
PRASANNA S GANDHI gandhi@me.iitb.ac.in



Recall

Four Bit Ring Counter

- Simplest counter: timing diagram?



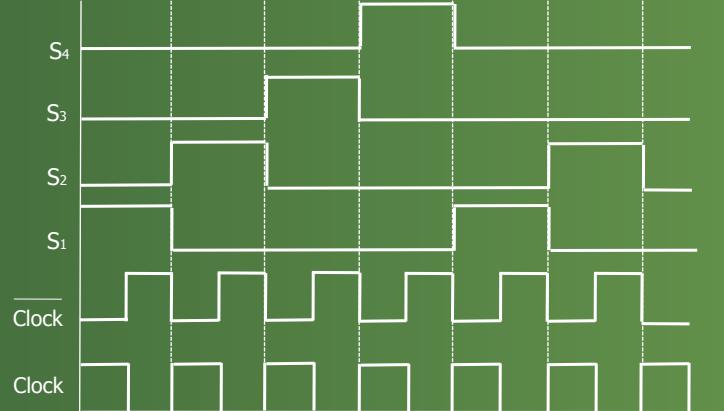
- How many states are getting represented? 1000, 0100, 0010, 0001
- Q: what if we start at 0000??
- Reset to 1000 needed to start.
- How to reset to 1000???

PRASANNA S GANDHI gandhi@me.iitb.ac.in

8



Timing diagram



PRASANNA S GANDHI gandhi@me.iitb.ac.in

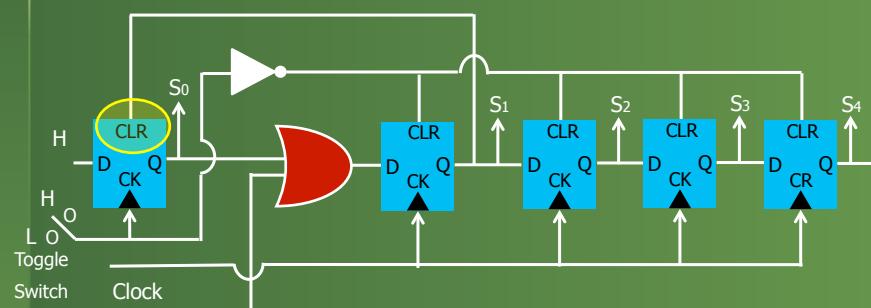
9



Another way to get reset at 1000

Circuit for generating sequence of pulses

Sequence starts on toggle of switch from low to high and continues



Notice: The clock of the first **positive edge triggered** D-FF is connected to toggle so operates only once at the start
Sequence continues as long as Toggle is held high

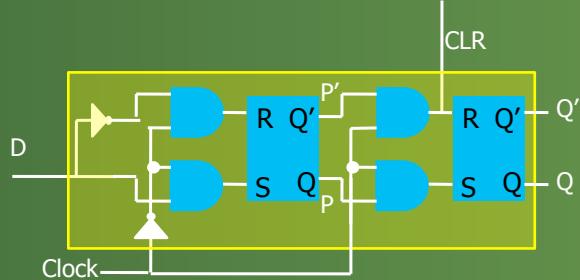
Timing diagram??

10



Positive Edge Triggered D-FF with CLR

- One more way (avoiding gates)



PRASANNA S GANDHI gandhi@me.iitb.ac.in

11



Operation of Data Transfer to Memory

- Q: How to use this sequence now to realize the operation of data transfer from external register to memory.
- See carefully again the timing diagram of operations we want with reference to timing diagram of the sequence we produced

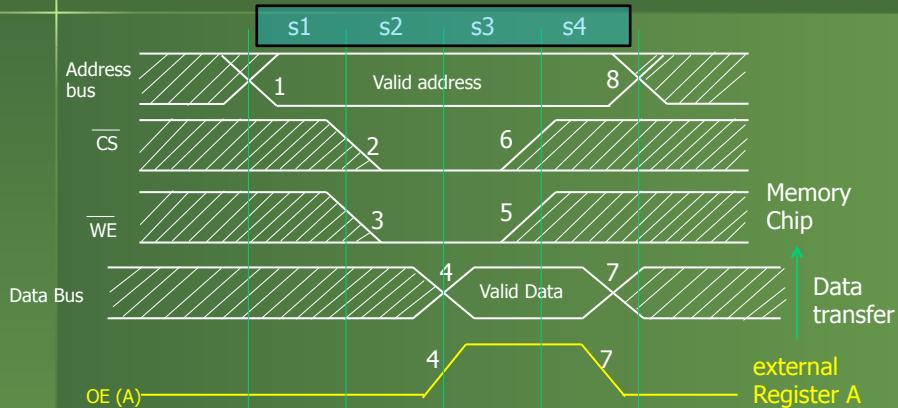
PRASANNA S GANDHI gandhi@me.iitb.ac.in

12



Timing diagram for writing data

to memory at location given by valid address from external register.



s1 is high in the interval indicated same with s2 s3 and s4

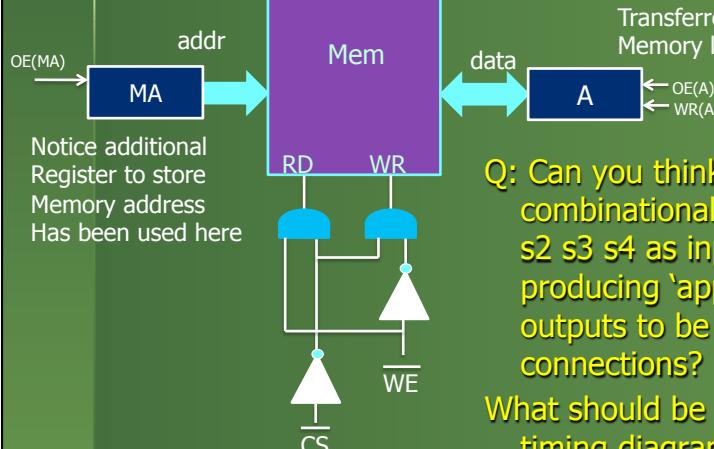
Q: can you think of combinational logic and hardware circuit to achieve this operation?

PRASANNA S GANDHI gandhi@me.iitb.ac.in

13



Physical connections



Notice additional Register to store Memory address Has been used here

This is external register From which data is to be Transferred to specified Memory location

Q: Can you think of combinational logic using s1 s2 s3 s4 as inputs and producing 'appropriate' outputs to be used in these connections?

What should be outputs? See timing diagram → CS, WE, OE (A) OE(MA)

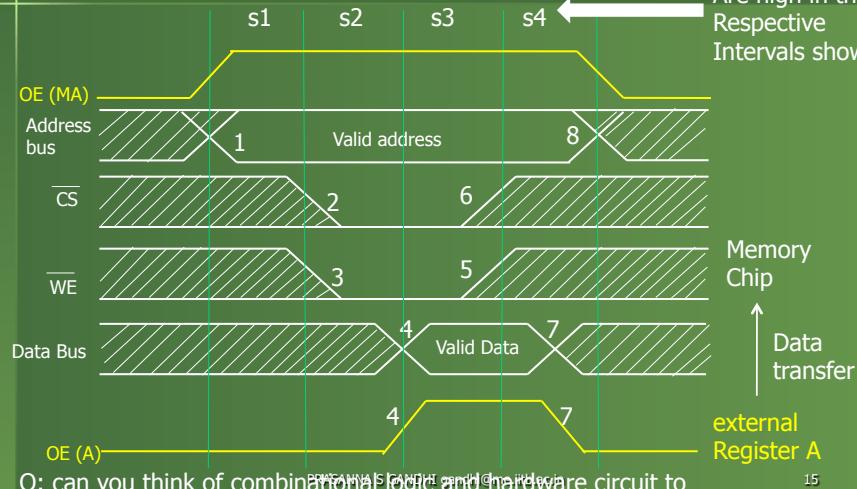
PRASANNA S GANDHI gandhi@me.iitb.ac.in

14



Timing diagram for write operation

This indicates
That s1 s2 s3 s4
Are high in the
Respective
Intervals shown

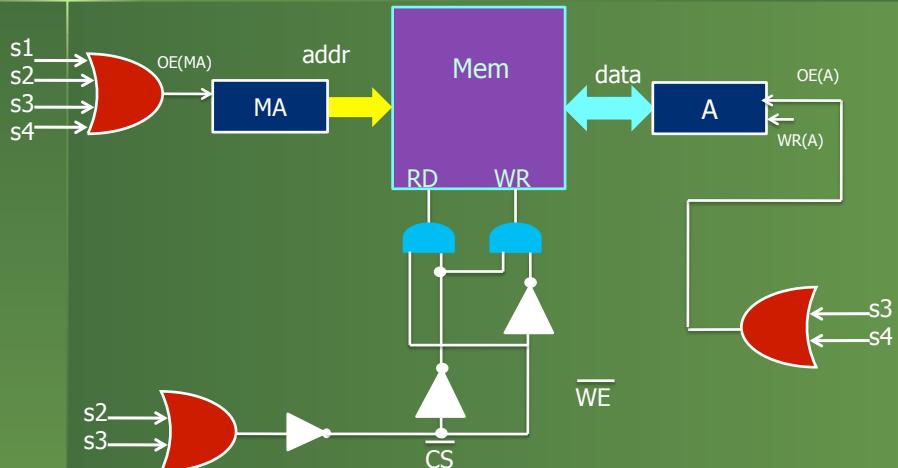


Q: can you think of combinational logic and hardware circuit to achieve this operation? Do not require K map to do this.. Isn't it?

15



Physical Connections



PRASANNA S GANDHI gandhi@me.iitb.ac.in

16



Basic Building Blocks

- Combinational logic:
 - Adder
 - Comparator
 - Decoder and encoder
 - Multiplexer and demux
 - And so on...
- Sequential logic:
 - RS and D-flip flops
 - Registers or basic memory element
 - Counters finite state machines (FSM): seq + combinational logic



How do we now work with these to get microprocessor?

- Use some temporary registers to bring in/out data from memory and do some operations with the data
- Operations can be executed using combinational logic circuits and output can be stored in some register or then to memory
- We further would need some kind of timing and control unit to carry out desired tasks
- Lets consider the following example



Primitive Microprocessor 1

- Now we will put together memory registers and more combinational logic to construct primitive microprocessor (purpose is to understand underlying philosophy)

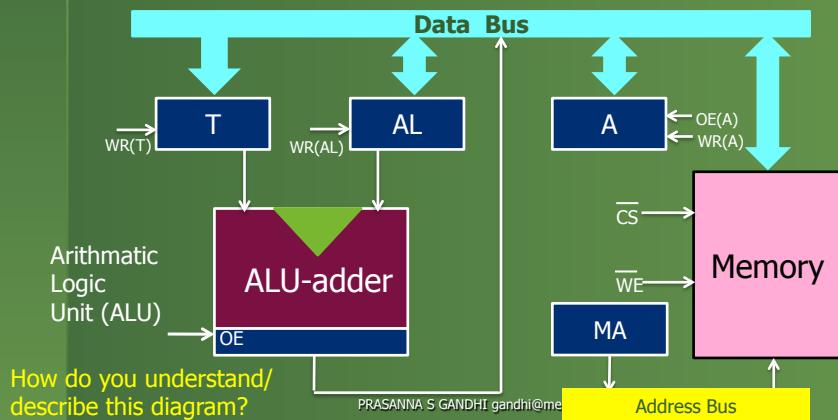
PRASANNA S GANDHI gandhi@me.iitb.ac.in

19



Primitive Microprocessor 1

- Consider the primitive muP architecture shown





Primitive Microprocessor 1

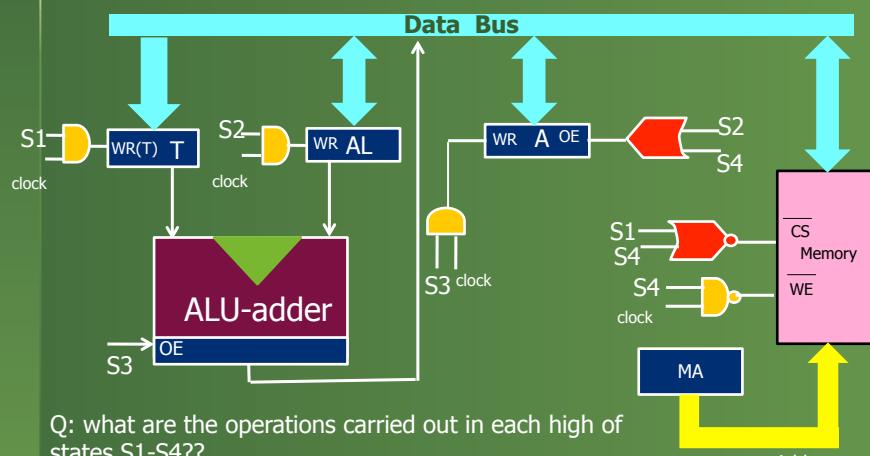
- Q: How do you understand/describe this diagram?
 - This may consist of two registers A and AL which are data read/write, and one write-only register T
 - ALU has only one operation addition. It adds contents of register T and AL and saves it in its register
 - It has memory address bus of which is connected to register MA
 - No specification has been given about size of memory and the size of the data
- Now suppose if the connections of four signals s1, s2, s3 and s4 are given below, guess what operations would be achieved?

PRASANNA S GANDHI gandhi@me.iitb.ac.in

21



Use states S1-S4 from previous example

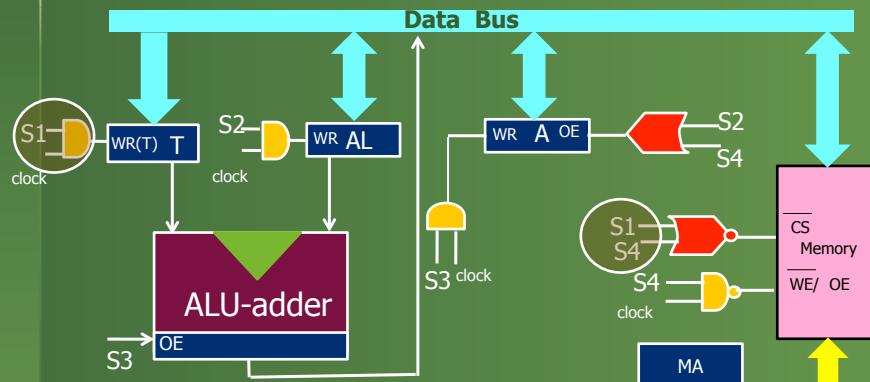


PRASANNA S GANDHI gandhi@me.iitb.ac.in

Address 22



Use states S1-S4 from previous example



Lets focus on part when **s1** is high

Observe: Contents of register **MA** will always be on the address bus

S1 goes high \rightarrow data at the location **MA** would be transferred to data bus

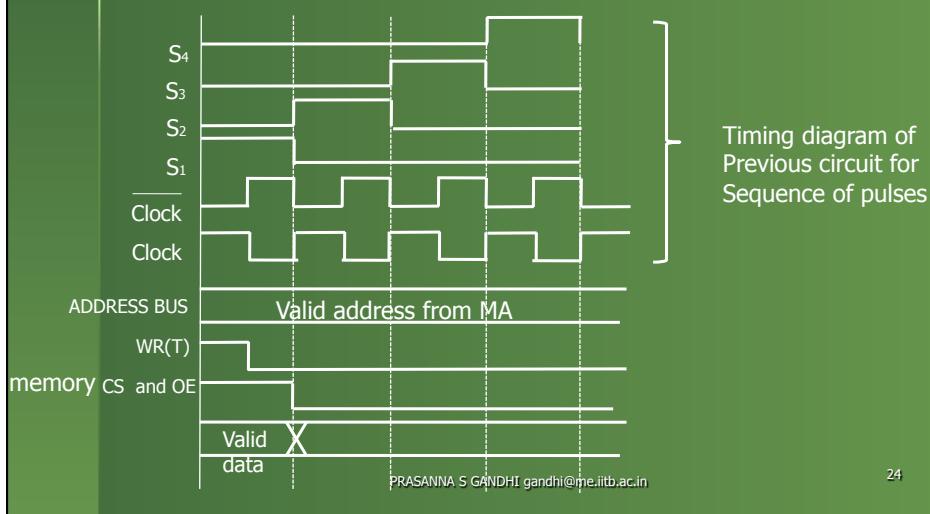
Further it would be written to register **T**

PRASANNA S GANDHI gandhi@me.iitb.ac.in

23

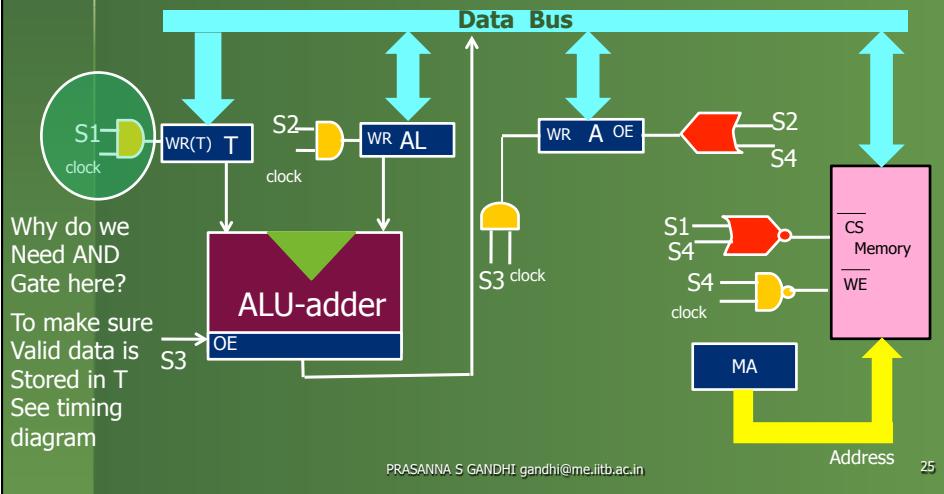


Timing diagram

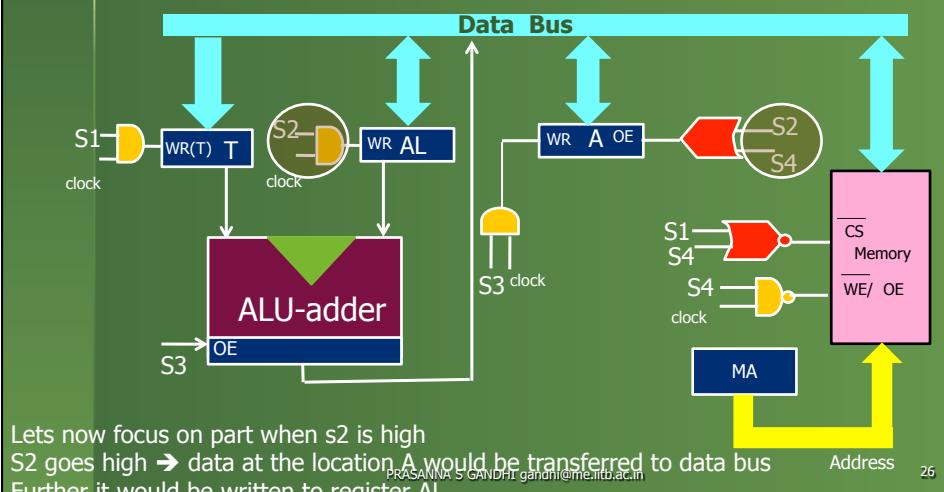




Use states S1-S4 from previous example

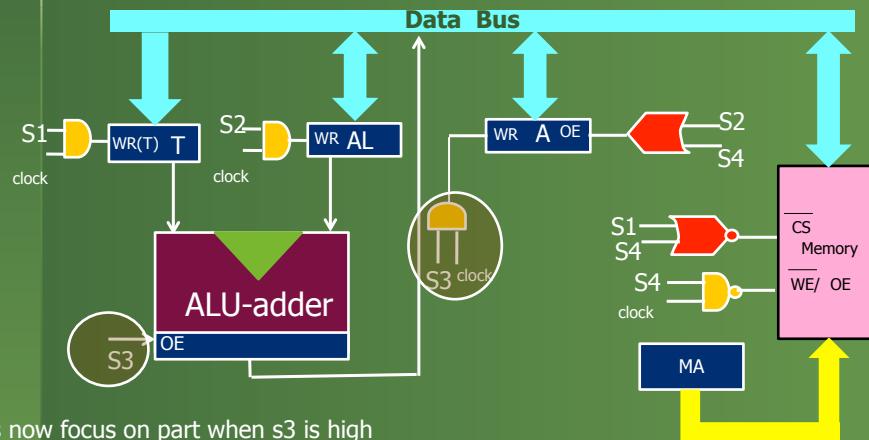


Use states S1-S4 from previous example





Use states S1-S4 from previous example

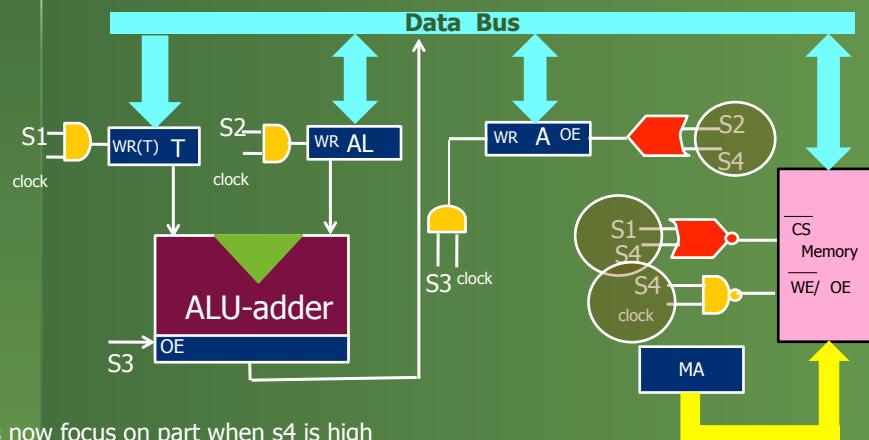


Lets now focus on part when s3 is high
S3 goes high → sum at the location ALU would be transferred to data bus
Further it would be written to register A

27



Use states S1-S4 from previous example



Lets now focus on part when s4 is high
S4 goes high → sum at the location A would be transferred to data bus
Further it would be written to register in the memory location pointed by address in MA

28



Summary of

Operation achieved

- Say in state S1 ($s_1=1, s_2=0, s_3=0, s_4=0$): $cs' = 0$ chip will be selected, contents MA will be on address bus memory (always), read state → data at location MA would be on data bus and would be transferred to register T
- State S2 (0100): Data from register A will be transferred to register AL
- State S3 (0010): Sum (calculated by adder in ALU) will be transferred to A
- State S4 (0001): data from A to memory location

PRASANNA S GANDHI gandhi@me.iitb.ac.in

29



Summary of

Operation achieved

- For example if initial contents of A = x and initial contents of memory at addressed location = y.
- In S1, S2 state: Contents of T will be y and contents of AL will be x. Addition $z = x+y$ will be in ALU.
- In S3,S4 : contents of A = z content of memory at addressed location = z
- Next cycle $z + z = 2z$; contents of A = $2z$, contents of memory at addressed location = $2z$
- Next cycle $2z+2z = 4z$; and same thing repeats

PRASANNA S GANDHI gandhi@me.iitb.ac.in

30



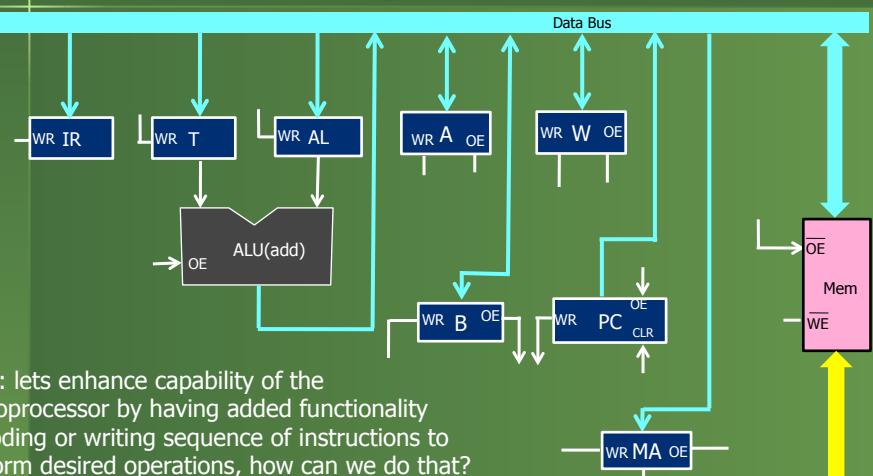
Operation achieved

- Not very meaningful
- 'Timing and control' gives only one possibility
- Only one command or instruction or sequence of states (its like a one line c code, not meaningful isn't it?)
- Four 'machine (clock) cycles or 'T' states' in this command
- We need more meaningful operations to be done
- How can we do that??
- Lets add more registers to our primitive microprocessor and then define some operations

PRASANNA S GANDHI gandhi@me.iitb.ac.in

31

Primitive Microprocessor 2



Next: lets enhance capability of the microprocessor by having added functionality of coding or writing sequence of instructions to Perform desired operations, how can we do that?

PRASANNA S GANDHI gandhi@me.iitb.ac.in

32



Primitive Microprocessor 2

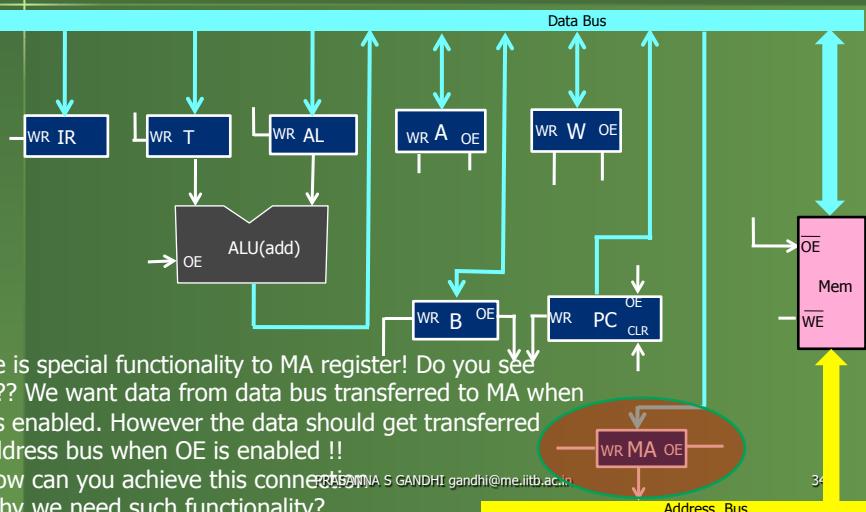
- Q: How do you understand/describe this diagram?
 - Description same as Primitive MuP 1 for common registers
 - Notice there are additional registers: IR, W, B, and PC (program counter). W, B are general purpose registers with same read write functionality as A.
 - PC is special purpose register to keep track of sequence of execution of operations. We would like to have this sequence pre-stored and changeable or programmable. (Its similar to writing c code commands one after other).
 - IR is instruction register which allows us to write instructions (similar to each command in C code). Depending on instruction different operations would be carried out.

PRASANNA S GANDHI gandhi@me.iitb.ac.in

33



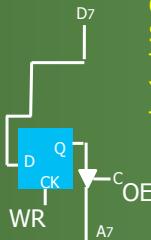
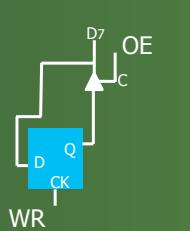
Primitive Microprocessor 2





MA Register Construction

- Consider one bit/unit in normal registers



Q: can OE be high at the Same time when WR high for This case?
YES Its possible. Think why?
The loop is broken

Same unit can be repeated for say 8 bit data bus and 8 bit address bus with all OEs connected together and all WRs connected together

- How one bit for MA register would look like? What modifications are needed to be carried out? See above

PRASANNA S GANDHI gandhi@me.iitb.ac.in

35



Enhanced Operations: Coding

- Q: We would like to have a facility in which we write some code or program (sequence of instructions in serial manner) in the memory and then it should get executed. Lets say we have some instructions written at memory locations say 10, 11, 12 and so on. We would like to have these instructions brought into register IR sequentially. How would you propose to do it using register PC (program counter)???
- First program counter (PC) register should hold number 10 (memory location where first instruction is stored), then what?? How do we fetch this instruction and place it in IR??

PRASANNA S GANDHI gandhi@me.iitb.ac.in

36



Enhanced Operations: Coding

- To get the first instruction (written at location 10 in memory) from memory number in PC should be moved to register MA so that register at this location (10) is selected for read (OE) operation. This can be briefly represented as $MA \leftarrow PC$
- Next step now is to transfer contents of this register in memory to data bus by enabling OE (mem).
- Next the data on data bus should be written to register IR. Both of these can be briefly represented as $IR \leftarrow M[MA]$
- Then we need to increment PC register by 1 so as to keep it ready for getting the next instruction $PC \leftarrow PC + 1$

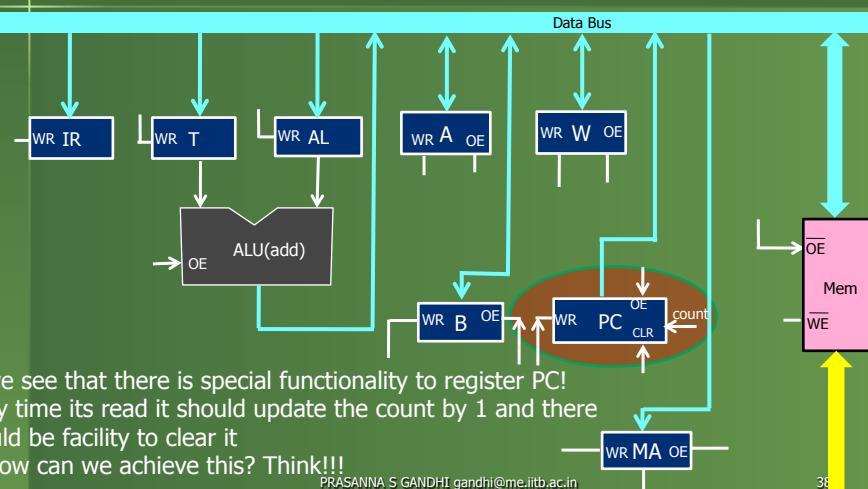
PRASANNA S GANDHI gandhi@me.iitb.ac.in

37



Functionality/Notion of programming added

Primitive Microprocessor 2



So we see that there is special functionality to register PC!
Every time its read it should update the count by 1 and there
Should be facility to clear it
Q: How can we achieve this? Think!!!

PRASANNA S GANDHI gandhi@me.iitb.ac.in

38



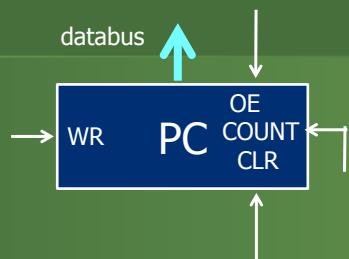
Program Counter Details

- How can we achieve program counter to function in a way described?
- There can be multiple possibilities again!
 - 1. Use the memory register and adder to achieve the job (this however cannot be achieved in one clock cycle)
 - 2. Use actually the counter we had seen in previous class
- Lets say we use second option and assume for sake of discussion and understanding fundamentals that its counting only 8 states (3 bits) → we take a counter previously designed and provide additional facility needed!!! **Think How would you achieve this??**

PRASANNA S GANDHI gandhi@me.iitb.ac.in 39



Program Counter Details

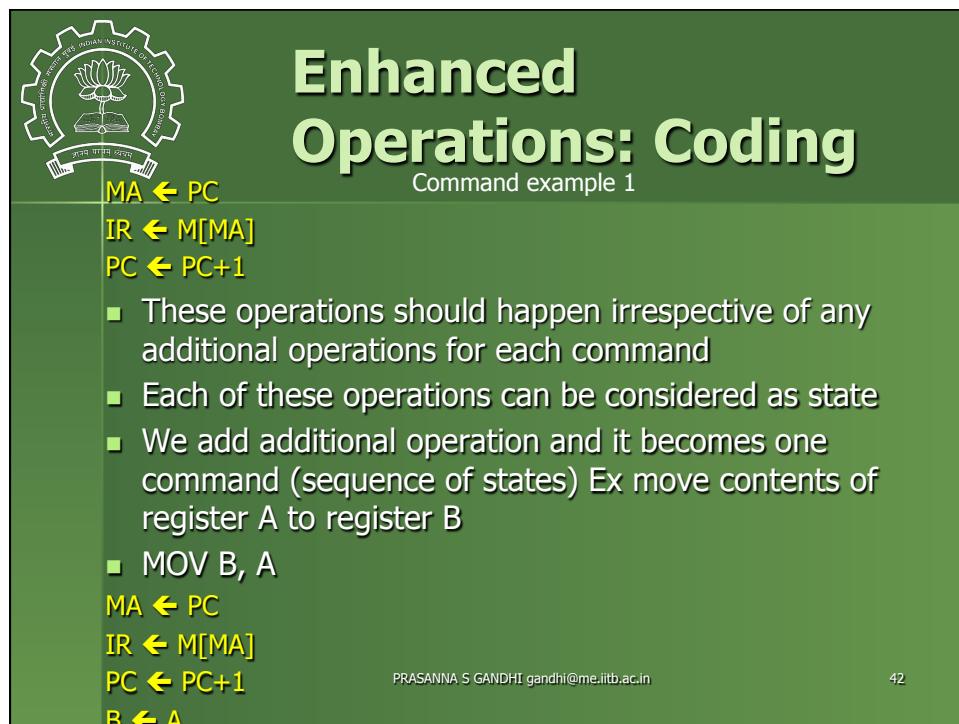
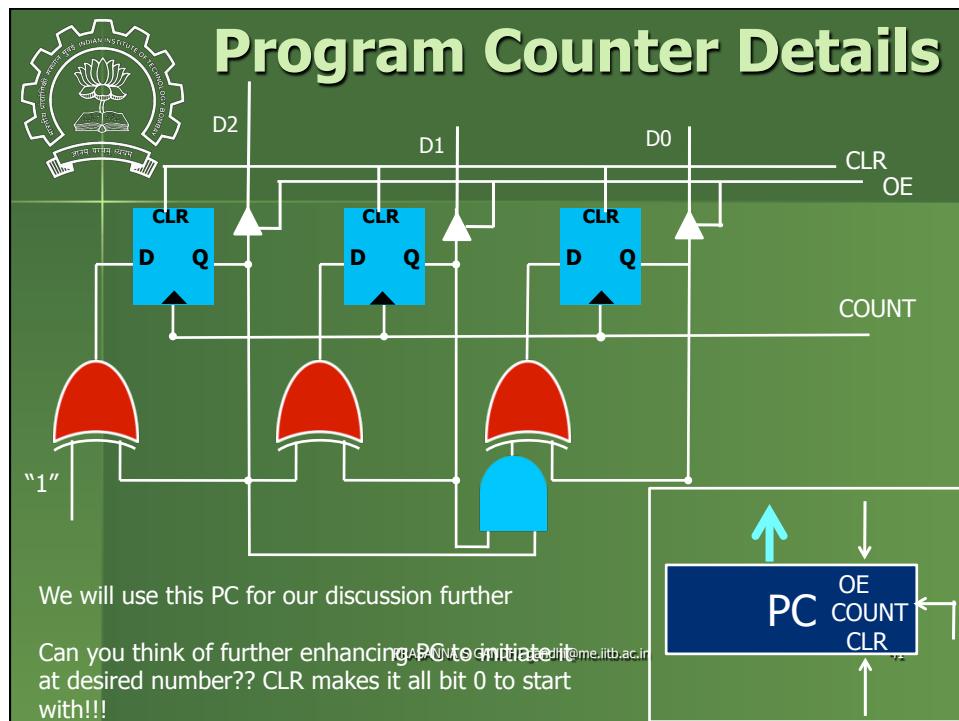


Only when the pin COUNT sees a pulse count is updated. **How will we develop such circuit?? Think!!!**

So what we want is to give control of update (clock input to Dffs of the counter circuit) to be considered as an external input and be connected to COUNT!! Isnt it?
See circuit in the next slide

PRASANNA S GANDHI gandhi@me.iitb.ac.in

40





Enhanced Operations: Coding

Command example 2

$$IR \leftarrow M[MA]$$

$$PC \leftarrow PC+1$$

- Another command may consist of some different operation (above 3 are going to be common to all commands). Example say add of two numbers stored in registers A and B and store the addition in register A. By looking at the microprocessor architecture can you write down sequence of operations (states)?

$$MA \leftarrow PC$$

$$IR \leftarrow M[MA]$$

$$PC \leftarrow PC+1$$

$$AL \leftarrow A$$

$$T \leftarrow B$$

$$A \leftarrow AL+T$$

Funda: To qualify what you chose as an operation or state It should be achievable in one machine (clock) cycle .

For example by looking at the microprocessor construction $A \leftarrow A + B$ cannot qualify as single operation. Think why? You cannot perform this in one clock cycle?

Q: why we need this funda at all? It is to finally develop sequential circuitary to achieve the command result?

PRASANNA S GANDHI gandhi@me.iitb.ac.in

43



Summary of Other commands

Operation to be carried out

Notice different Feature of instruction
3. Instruction is 2 part instruction. First part is actual command and the 2nd part holds more information needed to execute: address of memory location from where data is to be fetched to A

No	'T' States Sequence	Microinstructions executed in each S	Mnemonic	Code	Result
1	S1 → S2 → S3	S1 (MA) ← (PC) S2 (IR) ← M[MA]; (PC) ← (PC)+1 S3 (B) ← (A)	MOV B,A	00	Contents of A register is moved to be B register
2	S1 → S2 → S4 → S5 → S6	S1 (MA) ← (PC) S2 (IR) ← M[MA]; (PC) ← (PC)+1 S4 (AL) ← (A) S5 (T) ← (B) S6 (A) ← (T)+(AL)	ADD B	01	The Contents of B is added to the contents of A and placed result in A
3	S1 → S2 → S7 → S8 → S9 → S10	S1 (MA) ← (PC) S2 (IR) ← M[MA]; (PC) ← (PC)+1 S7 (MA) ← (PC) S8 (W) ← M[MA]; (PC) ← (PC)+1 S9 (MA) ← (W) S10 (A) ← M[MA]	LDA addr	10	Command with 2 parts: 2 memory location to store The Contents of the location whose address is given by the second byte after the opcode is moved to the register A
4	S1 → S2 → S7 → S8 → S9 → S11	S1 (MA) ← (PC) S2 (IR) ← M[MA]; (PC) ← (PC)+1 S7 (MA) ← (PC) S8 (W) ← M[MA]; (PC) ← (PC)+1 S9 (MA) ← (W) S11 M[MA] ← (A)	STA addr	11	Contents of register A are moved to the location in memory whose address is given by the second byte after the opcode

44



Based on four commands consisting of various operations
All possible states are collected together

Summary of state definitions

STATE/ Operation	Data transfer operation
S0	(PC) \leftarrow 0
S1	(MA) \leftarrow (PC)
S2	(IR) \leftarrow M[MA] ; (PC) = (PC)+1
S3	(B) \leftarrow (A)
S4	(AL) \leftarrow (A)
S5	(T) \leftarrow (B)
S6	(A) \leftarrow (T) + (AL)
S7	(MA) \leftarrow (PC)
S8	(W) \leftarrow M[MA]; (PC) \leftarrow (PC)+1
S9	(MA) \leftarrow (W)
S10	(A) \leftarrow M[MA]
S11	M[MA] \leftarrow A

S0 is added to initiate PC
Others are from previous

Note: some of the states are common to various commands.

Q: Are number of Commands and number of states related? Think!

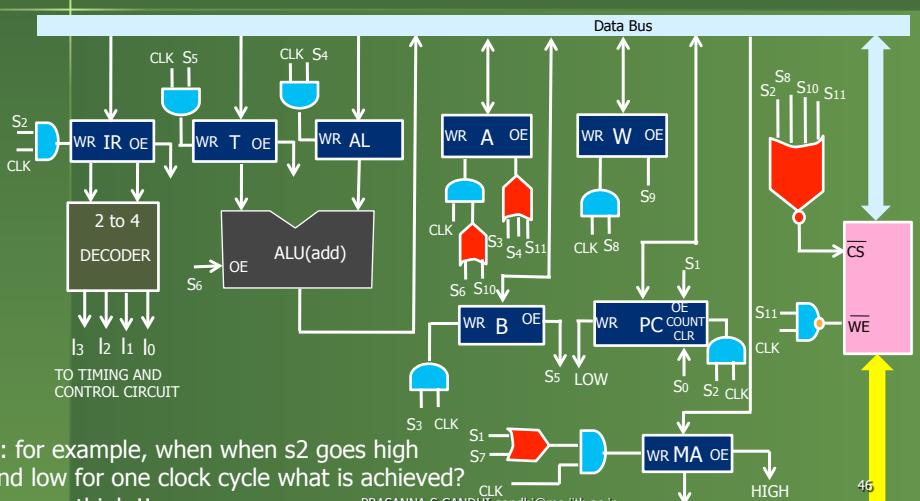
Q: how to use these definitions in our primitive mup 2 to achieve operations??

Say when signal s1 is high operation (MA) \leftarrow (PC) should be carried out.

45



Connections of Signals to Achieve the Tasks



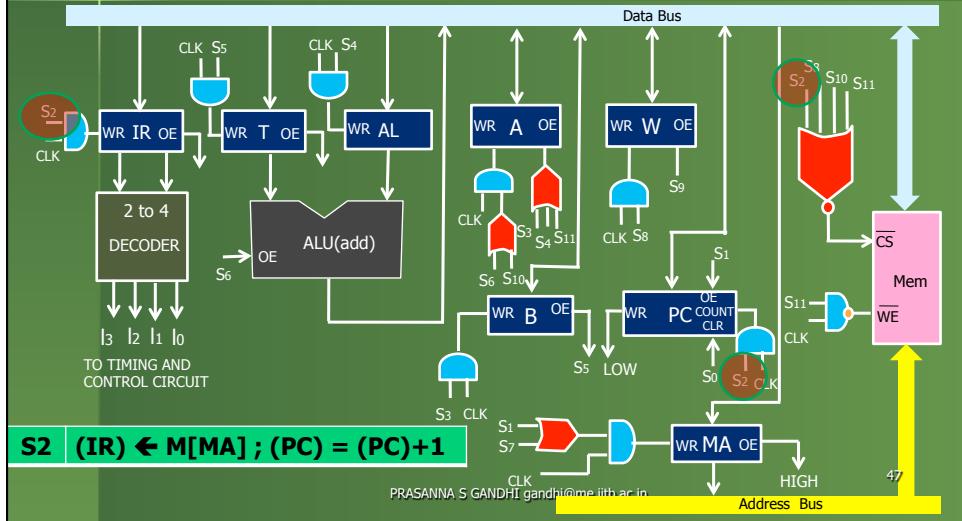
PRASANNA S GANDHI gandhi@iitb.ac.in

46



Example check state s2

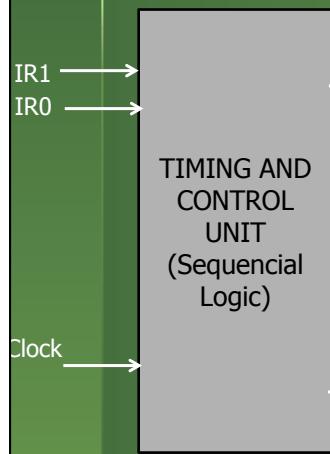
Connections of Signals to Achieve the Tasks



- Q: Now how to loop through each of the different loops seen before



Timing and Control Unit



- Total states : 12
- Instead of having single loop we will have four loops
- Loop to be executed is governed by combination of IR0 and IR1
- Use decoder to generate I0, I1, I2, I3 based on IR0 and IR1

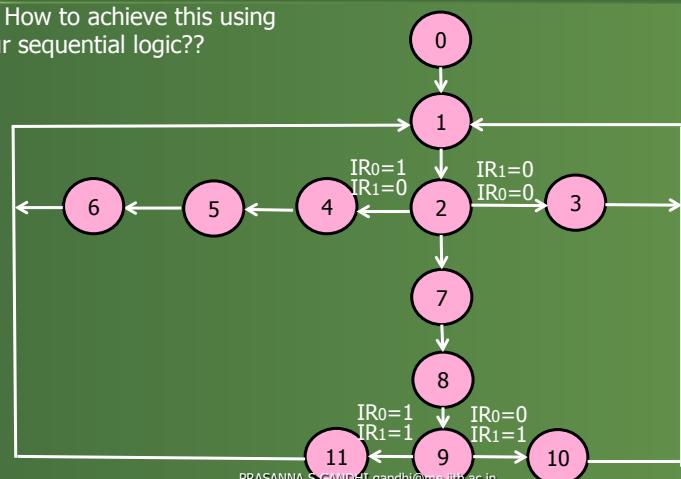
PRASANNA S GANDHI gandhi@me.iitb.ac.in

49



State diagram

Q: How to achieve this using Our sequential logic??

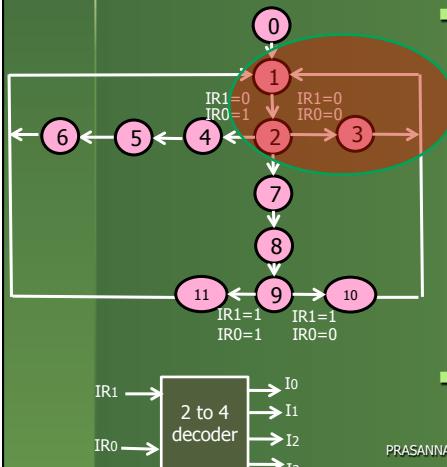


PRASANNA S GANDHI gandhi@me.iitb.ac.in

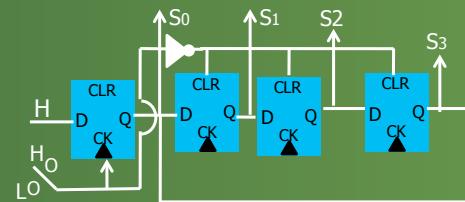
50



Timing and Control Unit



- Consider one loop how will you achieve this without control of IR0 and IR1?? Same as before see below!

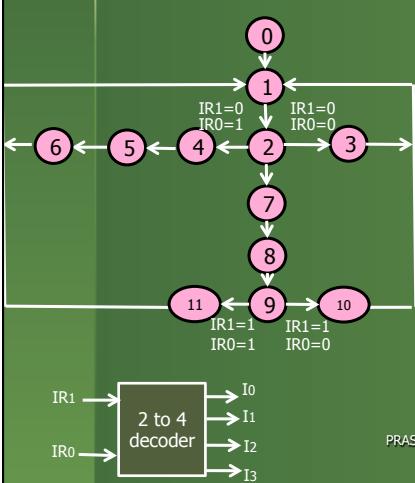


- However we want now additional control of code 00. We first use decoder to get signals I0, I1, I2, I3 corresponding to four instructions

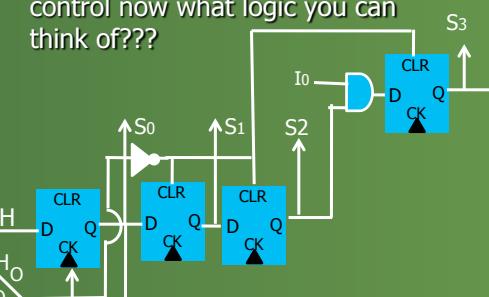
PRASANNA S GANDHI gandhi@me.iitb.ac.in



Timing and Control Unit



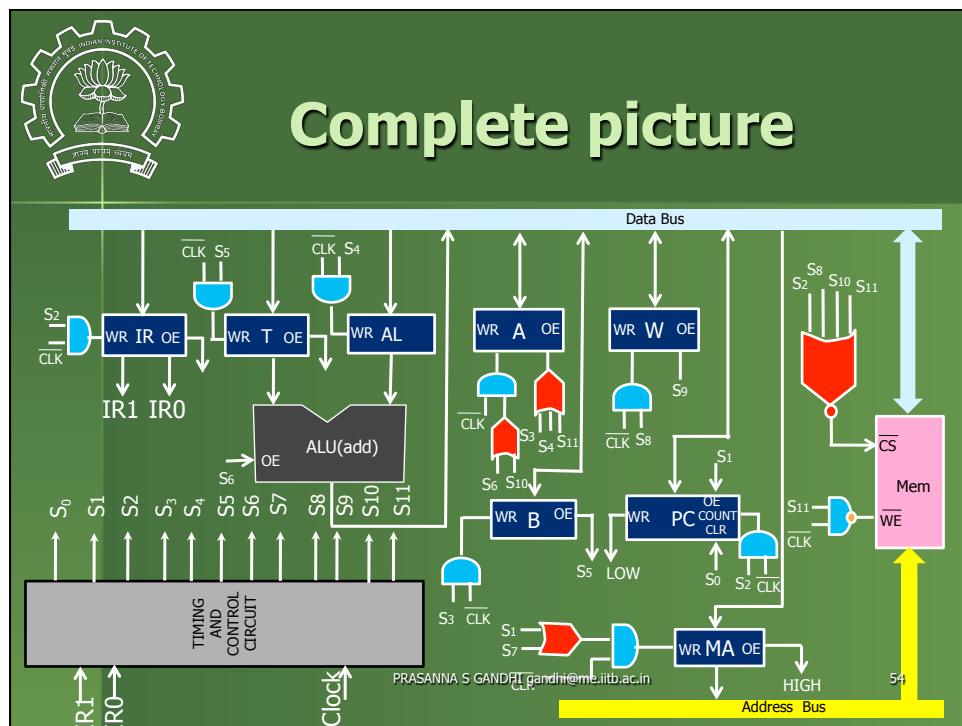
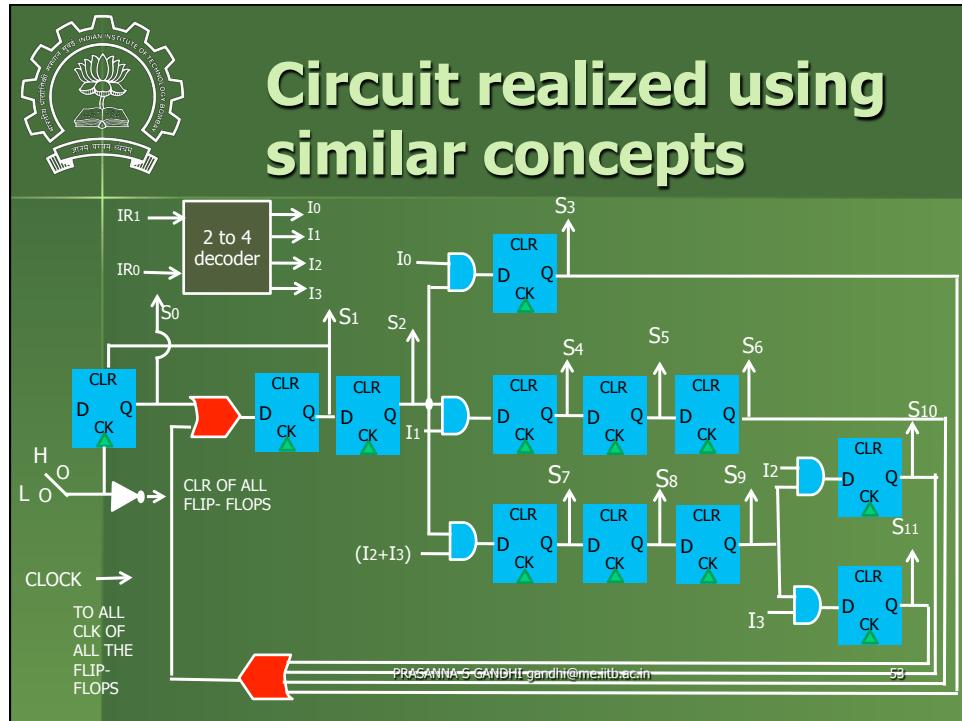
- To use these signals to have control now what logic you can think of???



- Can you think in the same way as this for getting the circuits for other loops

PRASANNA S GANDHI gandhi@me.iitb.ac.in

52





Summary

- S1 and S2 are common to all paths: read memory location pointed by PC and place its contents in IR register and increment PC
- Four different paths can be executed by placing appropriate “code” in the IR register
- The sequence of codes to be executed is saved in memory (repeat possible) and fetched from there in every “path” by S1 and S2 commands
- We have designed a microcontroller having 4 instructions (codes 00, 01, 02, 03) executing

PRASANNA S GANDHI gandhi@me.iitb.ac.in

55



Summary

- Each of these group of instructions (executed for each Sx in one clock cycle) is called micro-instruction
- An instruction say **MOV B,A** consists of several micro-instructions (3 in this case)
- When muc is put on it first initializes PC to 0 (state S0) then goes through instructions S1 and S2. Thus the execution of instructions stored in memory locations from 0th address begins one by one.
- Thus desired program: “**in machine language**” can be developed

PRASANNA S GANDHI gandhi@me.iitb.ac.in

56



Summary

The program can be written in 'mnemonics' as

LDA 30H ← H here signifies hex:

MOV B,A Do you recall what this does?

LDA 31H :

ADD B

STA 32H

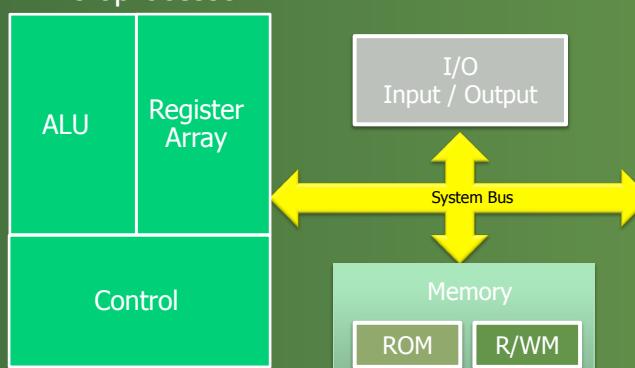
- Can you see what this program would do in terms of commands developed before
 - load A with contents of memory location 30H
 - Move contents of A to register B
 - Load A with contents of memory location 31H
 - Add contents of A to contents of B and store addition in A
 - Store contents of A to memory location 32H
- Easier to understand and develop bigger programs, without getting into details of hardware operations of making pins high⁵⁷ and low

PRASANNA S GANDHI gandhi@me.iitb.ac.in



Microprocessor system: basic architecture

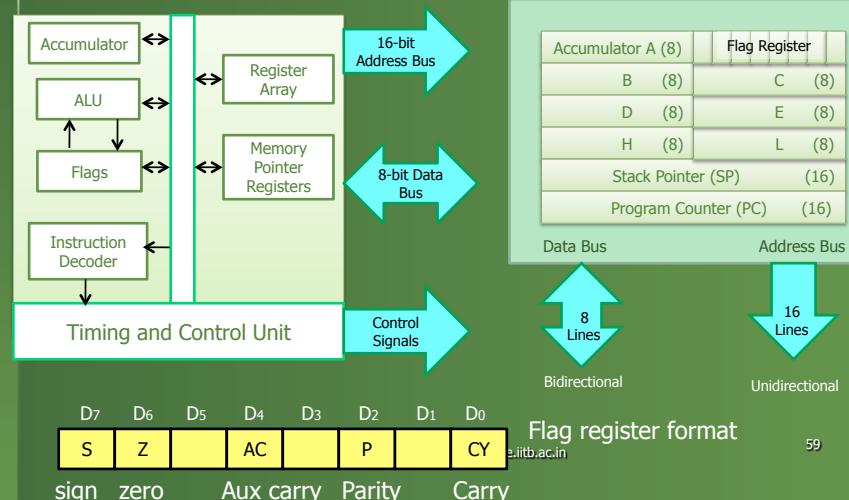
Microprocessor





Example: 8085 mup

ARCHITECTURE



Registers and their use

- Accumulator (A): part of ALU. Used to store 8 bit data and to perform arithmetic and logical operations. The result of operation is stored in A
- Flags: 5 FFs set or reset depending on data condition of result in A or other registers. Flags have critical role in decision making. Ex instruction JC (jump on carry) is implemented to change the sequence of program when CY flag is set.
- PC and SP-Stack pointer: 16 bit registers to hold memory addresses. PC used for sequencing execution of program in memory



Registers and their use

- SP is used for pointing to memory location in R/W memory called the stack. Beginning of the stack is defined by loading an address in the SP. Useful for subroutines. Do not worry how at this point! We will see later!!
- See datasheet for description and details of instructions etc. See book of Gaonkar for command reference

PRASANNA S GANDHI gandhi@me.iitb.ac.in

61



Thank You

PRASANNA S GANDHI gandhi@me.iitb.ac.in

62