Control Signals & Hardwired Control Unit

- How to execute various kinds of instructions using a single bus architecture?

ADD R1, R2 # R1= R1 + R2

Steps	Actions
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
2	Z_{out}, PC_{in}, Y_{in}, WMFC
3	MDR_{out}, IR_{in}
4	$R1_{out}$, Y_{in}
5	$R2_{out}$, SelectY, Add, Z_{in}
6	Z_{out}, $R1_{in}$, End

1. Output the value of the PC into the bus, then putting into MAR, activating READ signal, select4, add, after the addition i.e. the incremented value of PC is in Zin

2. The value from the Z bus goes to the PC and then we put it in Yin (Required for Branching) & wait for memory function complete

3. In the next step we do MDR out and IRin, i.e. the instruction after WMFC is available in MDR and sent to Irin. The instruction will now be decoded

4. One input comes through Y, so in tis case we consider the R1 coming from Y, i.e. outputting R1 to input of Y

5. Next we have to ouput R2 directly to the ALU and we selectY (containing value of R1), perform the addition and sent the result to Z

6. The result from Z is now sent to register R1in and finally end

ADD R1, LOCA # R1= R1 + MEM[LOCA]

Steps	Actions
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
2	Z_{out}, PC_{in}, Y_{in}, WMFC
3	MDR_{out}, IR_{in}
4	Address field of IR_{out}, MAR_{in}, Read
5	$R1_{out}$, Y_{in}, WMFC
6	MDR_{out}, SelectY, Add, Z_{in}
7	Z_{out}, $R1_{in}$, End

4. Address field of IRout is sent to MAR and then we proceed with reading.

5. While WMFC we can make R1 available at Y.

6. We do MDRout, Select Y and ADD and the Result is now in Z registers

LOAD R1, LOCA #**R1 = Mem[LOCA]**

Steps	Actions
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
2	Z_{out}, PC_{in}, Y_{in}, WMFC
3	MDR_{out}, IR_{in}
4	Address field of IR_{out}, MAR_{in}, Read
5	WMFC
6	MDR_{out}, $R1_{in}$, END

STORE LOCA, R1 # **Mem[LOCA] = R1**

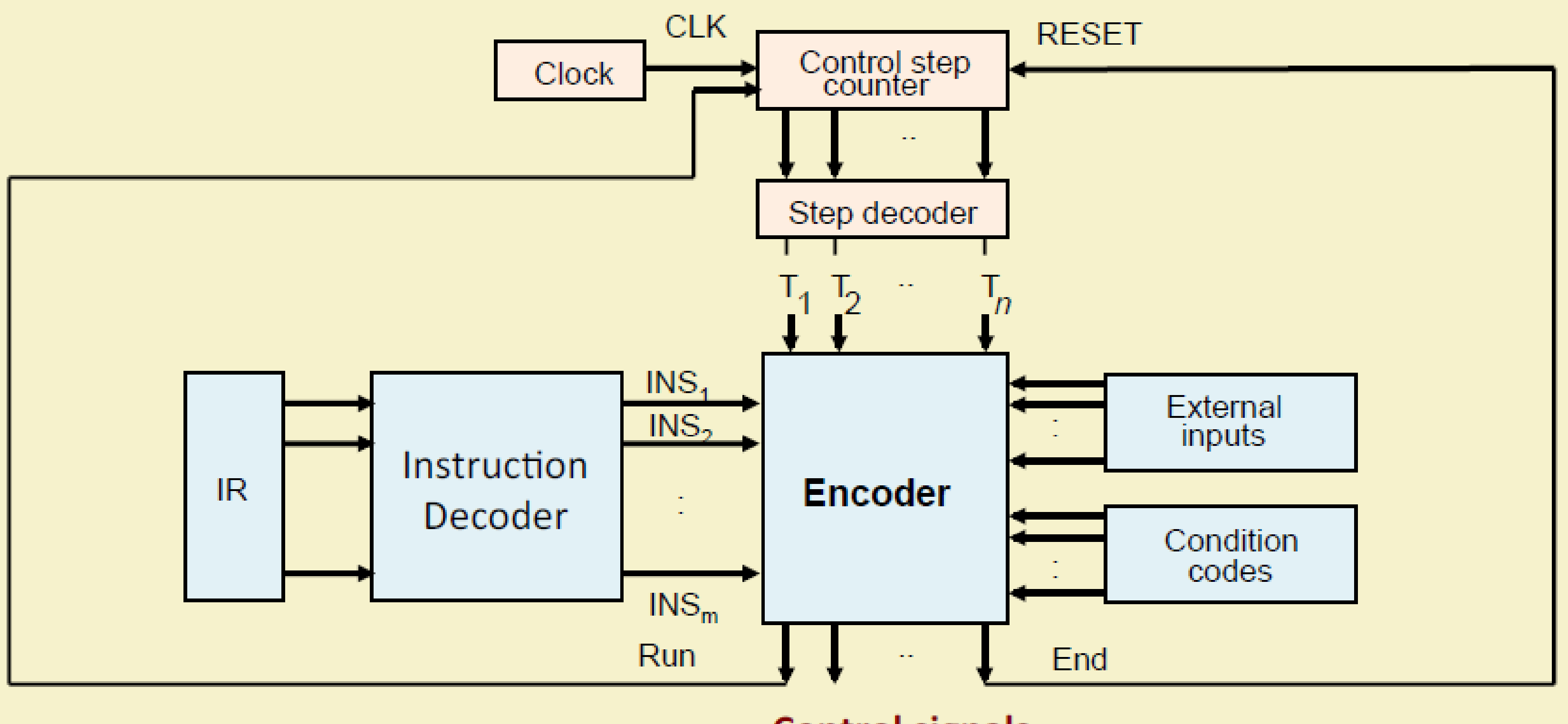
Steps	Actions
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
2	Z_{out}, PC_{in}, Y_{in}, WMFC
3	MDR_{out}, IR_{in}
4	Address field of IR_{out}, MAR_{in}
5	$R1_{out}$, MDR_{in}, Write
6	MDR_{outE}, WMFC, END

Methods to Design Control Unit

- The processor must generate proper sequence of control signals to execute the instructions

- Two designs approach:-
  - Hardwired control unit design
  - Microprogrammed control unit design

Hardwired Control unit



Control signals

1. Clock connected to a control step counter
2. Control step counter is connected to a step decoder to generate the steps
3. The encoder encoded the information received from the step decoder, the Instruction decoder and whether there are any external inputs and condition codes. The encoder generate the necessary control signals

Hardwired Control Unit Design

- Assumption:
  - Each step in this sequence is completed in one clock cycle.
- A counter is used to keep track of the time step.
- The control signals are determined by the following information:
  - Content of control step counter
  - Content of instruction register
  - Content of conditional code flags
  - External input signals such as MFC (Memory Function Complete)

- The encoder/decoder circuit is a combinational circuit which generates control signals depending on the inputs provided.
- The step decoder generates separate signal line for each step in the control sequence (T_1, T_2, T_3, etc.).
  - Depending on maximum steps required for an instruction, the step decoder is designed.
  - If a maximum of 10 steps are required, then a 4 x 16 step decoder is used.
- Among the total set of instructions, the instruction decoder is used to select one of them. (That particular line will be 1 and rest will be 0).
  - If a maximum of 100 instructions are present in the ISA then a 7 x 128 instruction decoder is used.

- At every clock cycle the RUN signal is used to increment the counter by one.
  - When RUN is 0 the counter stops counting.
  - This signal is needed when WMFC is issued.
- END signal starts a new instruction.
- It resets the control step counter to its starting value.
- The sequence of operations carried out by the control unit is determined by the wiring of the logic elements and hence it is named *hardwired*.
- This approach of control unit design is fast but limited to the complexity of instruction set that is implemented.

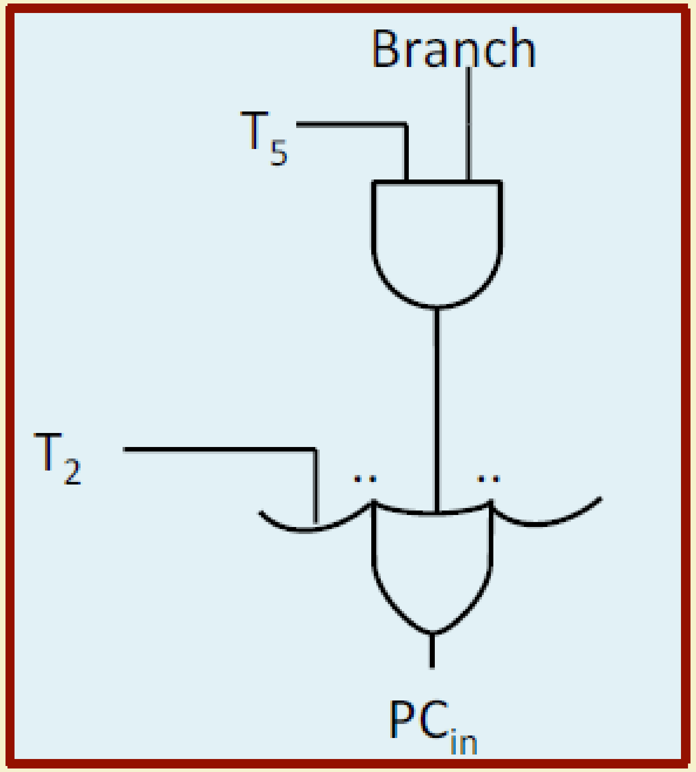
ADD R1, R2	
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
2	Z_{out}, PC_{in}, Y_{in}, WMFC
3	MDR_{out}, IR_{in}
4	$R1_{out}$, Y_{in}
5	$R2_{out}$, SelectY, Add, Z_{in}
6	Z_{out}, $R1_{in}$, End

ADD R1, LOCA	
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
2	Z_{out}, PC_{in}, Y_{in}, WMFC
3	MDR_{out}, IR_{in}
4	Address field of IR_{out}, MAR_{in}, Read
5	$R1_{out}$, Y_{in}, WMFC
6	MDR_{out}, SelectY, Add, Z_{in}
7	Z_{out}, $R1_{in}$, End

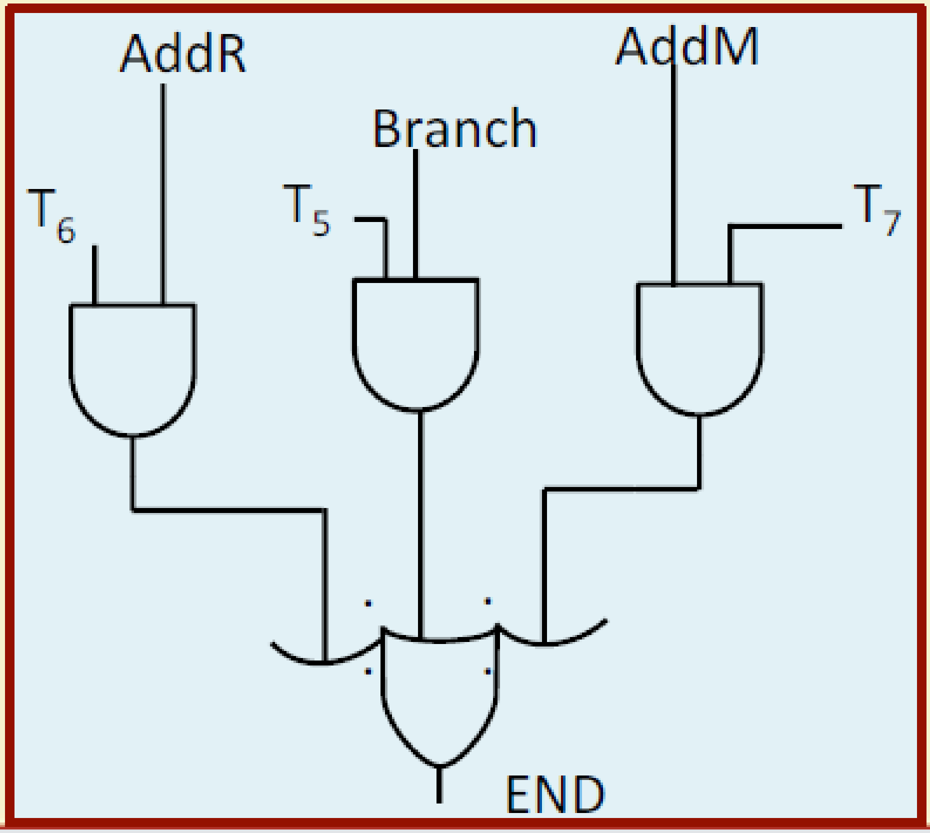
Branch Label	
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
2	Z_{out}, PC_{in}, Y_{in}, WMFC
3	MDR_{out}, IR_{in}
4	Offset-field-of IR_{out}, SelectY, Add, Z_{in}
5	Z_{out}, PC_{in}, End

Generation of PC_{in} and END

$PC_{in} = T_2 + T_5.Branch$



$END = T_6.ADDR + T_5.Branch + T_7.AddM$



Thank You