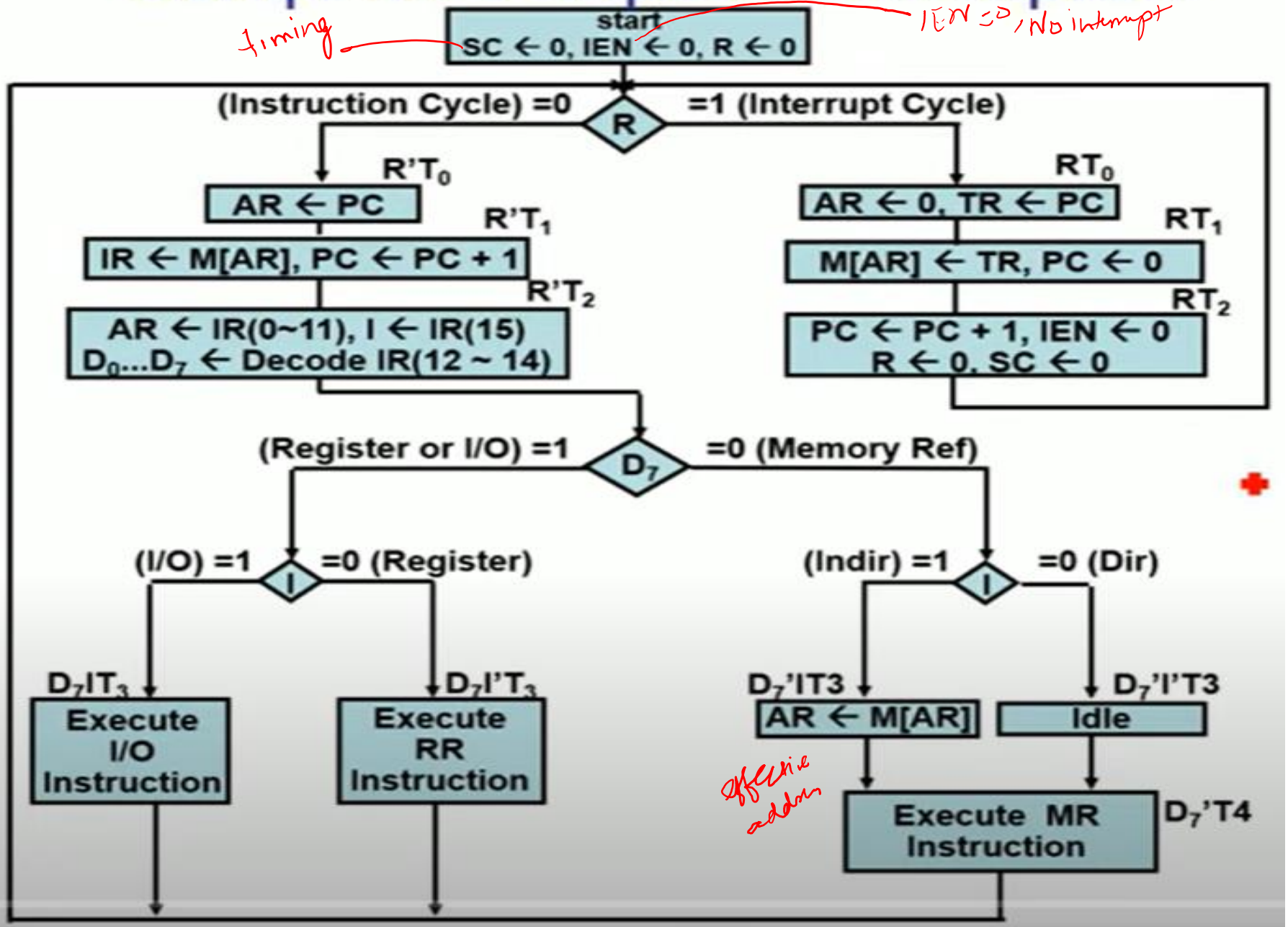
CHITKARA UNIVERSITY

HIMACHAL PRADESH - NAAC Accredited

Complete Computer Description

Lecture 15

Complete Computer Description



Fetch	R'T0:	AR ← PC
	R'T1:	IR ← M[AR], PC ← PC + 1
Decode	R'T2:	D0, ..., D7 ← Decode IR(12 ~ 14), AR ← IR(0 ~ 11), I ← IR(15)
Indirect	D7'IT3:	AR ← M[AR]
Interrupt:		
T0'T1'T2'(IEN)(FGI + FGO):		R ← 1
	RT0:	AR ← 0, TR ← PC
	RT1:	M[AR] ← TR, PC ← 0
	RT2:	PC ← PC + 1, IEN ← 0, R ← 0, SC ← 0
Memory-Reference:		
AND	D0T4:	DR ← M[AR]
	D0T5:	AC ← AC . DR, SC ← 0
ADD	D1T4:	DR ← M[AR]
	D1T5:	AC ← AC + DR, E ← Cout, SC ← 0
LDA	D2T4:	DR ← M[AR]
	D2T5:	AC ← DR, SC ← 0
STA	D3T4:	M[AR] ← AC, SC ← 0
BUN	D4T4:	PC ← AR, SC ← 0
BSA	D5T4:	M[AR] ← PC, AR ← AR + 1
	D5T5:	PC ← AR, SC ← 0
ISZ	D6T4:	DR ← M[AR]
	D6T5:	DR ← DR + 1
	D6T6:	M[AR] ← DR, if(DR=0) then (PC ← PC + 1), SC ← 0

Register-Reference:

	D7I'T3 = r	(Common to all register-reference instructions)
	IR(i) = Bi	(i = 0,1,2, ..., 11)
	r:	SC ← 0
CLA	rB11:	AC ← 0
CLE	rB10:	E ← 0
CMA	rB9:	AC ← AC'
CME	rB8:	E ← E'
CIR	rB7:	AC ← shr AC, AC(15) ← E, E ← AC(0)
CIL	rB6:	AC ← shl AC, AC(0) ← E, E ← AC(15)
INC	rB5:	AC ← AC + 1
SPA	rB4:	If(AC(15) =0) then (PC ← PC + 1)
SNA	rB3:	If(AC(15) =1) then (PC ← PC + 1)
SZA	rB2:	If(AC = 0) then (PC ← PC + 1)
SZE	rB1:	If(E=0) then (PC ← PC + 1)
HLT	rB0:	S ← 0

Input-Output:

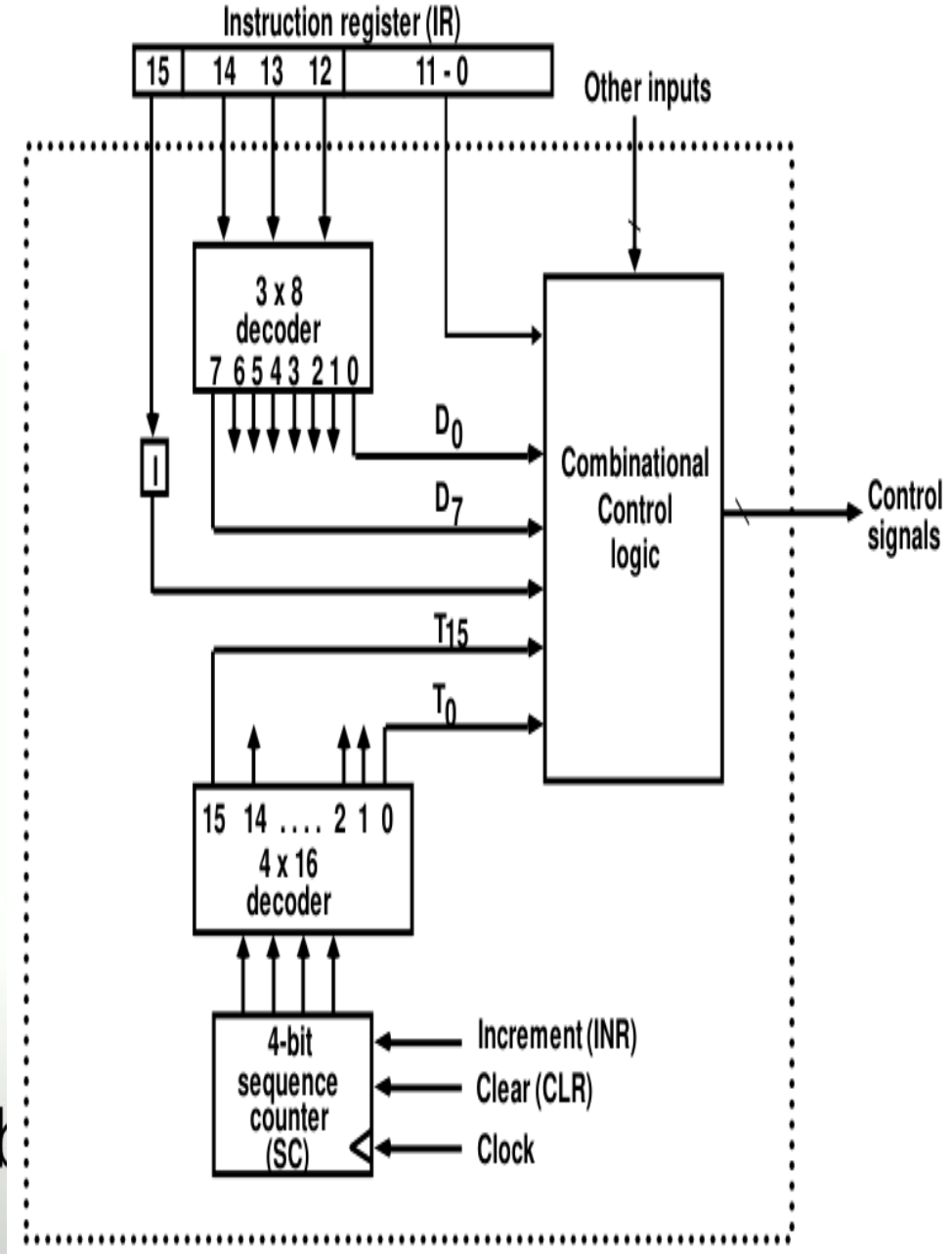
	D7IT3 = p	(Common to all input-output instructions)
	IR(i) = Bi	(i = 6,7,8,9,10,11)
	p:	SC ← 0
INP	pB11:	AC(0-7) ← INPR, FGI ← 0
OUT	pB10:	OUTR ← AC(0-7), FGO ← 0
SKI	pB9:	If(FGI=1) then (PC ← PC + 1)
SKO	pB8:	If(FGO=1) then (PC ← PC + 1)
ION	pB7:	IEN ← 1
IOF	pB6:	IEN ← 0

Design of Basic Computer

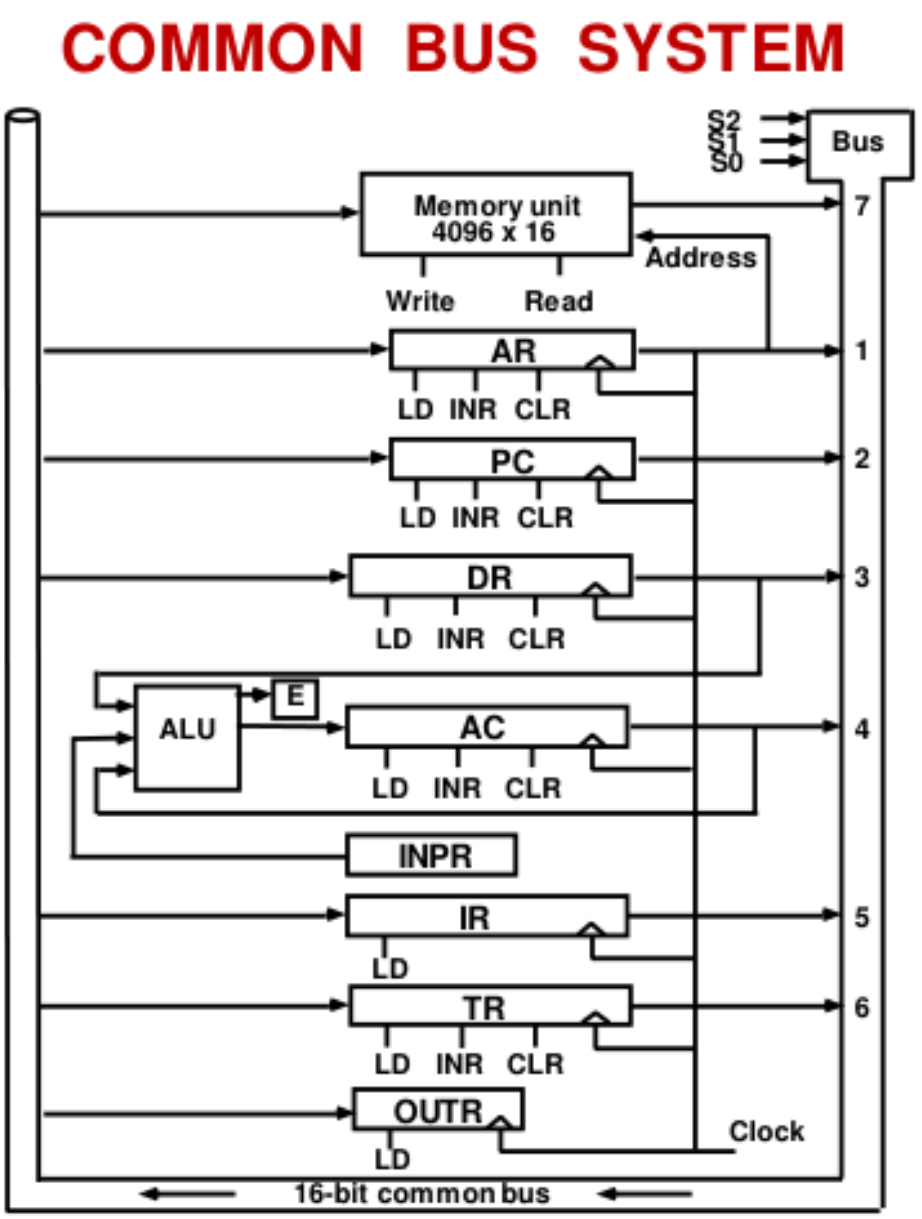
1. A memory unit: 4096 x 16.
2. Registers: AR, PC, DR, AC, IR, TR, OUTR, INPR, and SC
3. Flip-Flops (Status): I, S, E, R, IEN, FGI, and FGO
4. Decoders:
   1. a 3x8 Opcode decoder
   2. a 4x16 timing decoder
5. Common bus: 16 bits
6. Control logic gates
7. Adder and Logic circuit: Connected to AC

timing & control everything

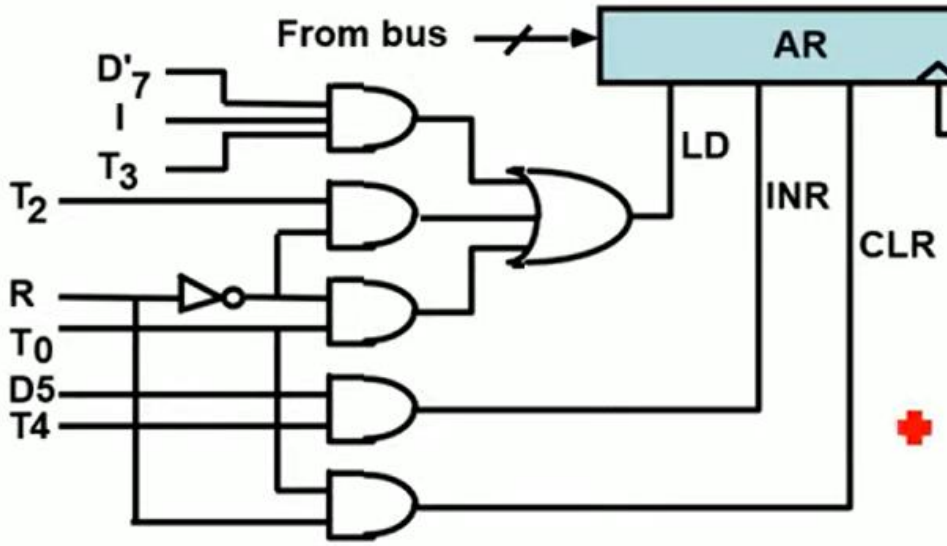
- The control logic gates are used to control:
  - Inputs of the nine registers
  - Read and Write inputs of memory
  - Set, Clear, or Complement inputs of the flops
  - S2, S1, S0 that select a register for the b
  - AC Adder and Logic circuit



- Control of registers and memory
  - The control inputs of the registers are LD (load), INR (increment), and CLR (clear)
  - To control AR We scan table 5-6 to find out all the statements that change the content c AR:
    - **R'T0:** $AR \leftarrow PC$ **LD(AR)**
    - **R'T2:** $AR \leftarrow IR(0\text{-}11)$ **LD(AR)**
    - **D'7IT3:** $AR \leftarrow M[AR]$ **LD(AR)**
    - **RT0:** $AR \leftarrow 0$ **CLR(AR)**
    - **D5T4:** $AR \leftarrow AR + 1$ **INR(AR)**



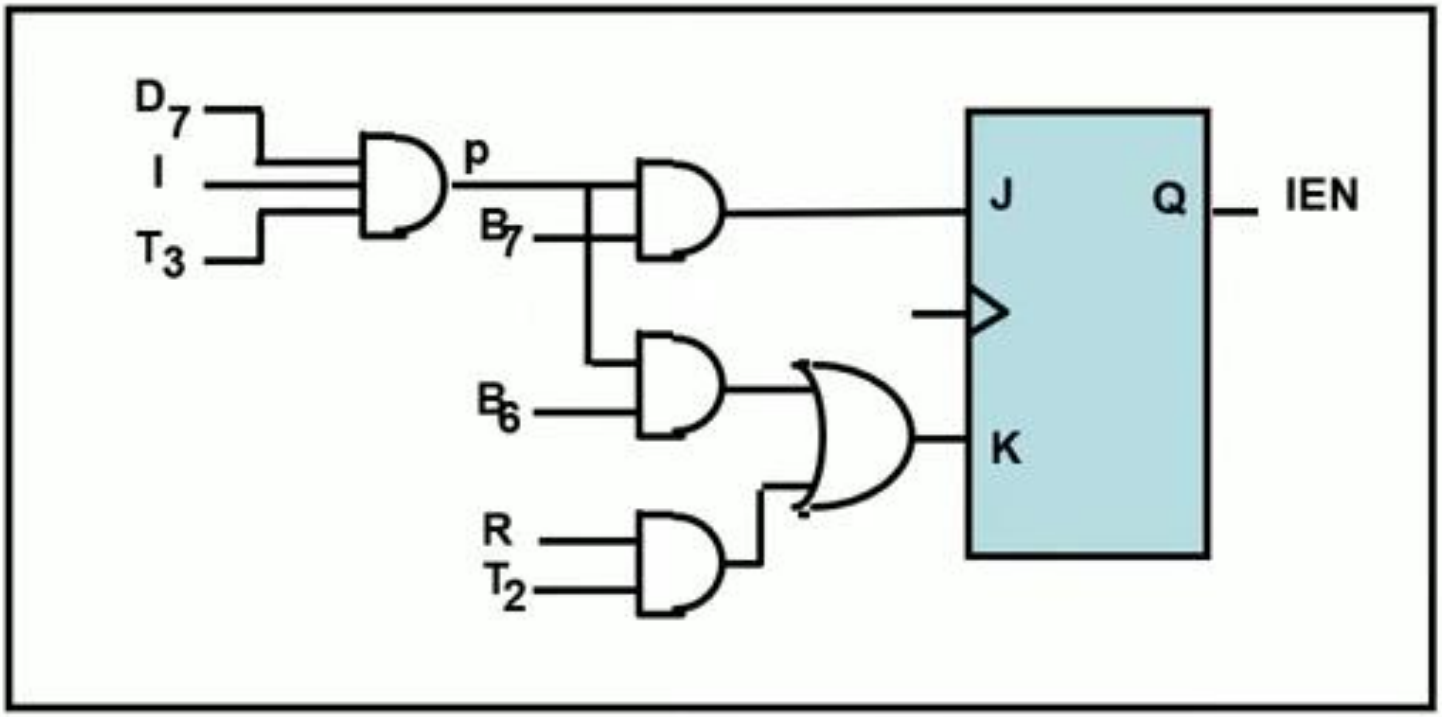
Control Gates associated with AR



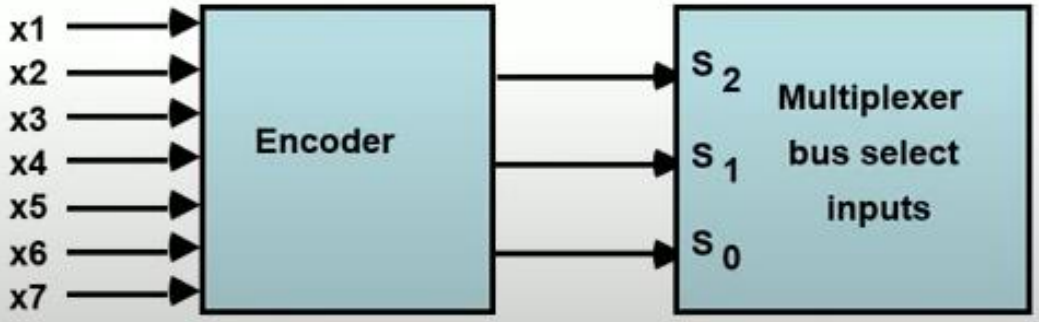
R'T0:	AR ← PC	LD(AR)
R'T2:	AR ← IR(0-11)	LD(AR)
D'7IT3:	AR ← M[AR]	LD(AR)
RT0:	AR ← 0	CLR(AR)
D5T4:	AR ← AR + 1	INR(AR)

- To control the Read input of the memory we scan the table again to get these:
  - D_0T_4: DR ← M[AR]
  - D_1T_4: DR ← M[AR]
  - D_2T_4: DR ← M[AR]
  - D_6T_4: DR ← M[AR]
  - $D_7'IT_3$: AR ← M[AR]
  - $R'T_1$: IR ← M[AR]
- → $Read = R'T_1 + D_7'IT_3 + (D_0 + D_1 + D_2 + D_6)T_4$

- Control of Single Flip-flops (IEN for example)
  - **pB7: IEN ← 1 (I/O Instruction)**
  - **pB6: IEN ← 0 (I/O Instruction)**
  - **RT2: IEN ← 0 (Interrupt)**
    - **where p = D7IT3 (Input/Output Instruction)**
  - If we use a JK flip-flop for IEN, the control gate logic will be as shown in the following slide:


D7
I
T3
P
B7
B6
R
T2
J
Q
IEN
K

- Control of Common bus is accomplished by placing an encoder at the inputs of the bus selection logic and implementing the logic for each encoder input



- To select AR on the bus then x_1 must be 1. This is happen when:
  - **D_4T_4: PC ← AR**
  - **D_5T_5: PC ← AR**
- $\Rightarrow x_1 = D_4T_4 + D_5T_5$

x1	x2	x3	x4	x5	x6	x7	S2	S1	S0	selected register
0	0	0	0	0	0	0	0	0	0	none
1	0	0	0	0	0	0	0	0	1	AR
0	1	0	0	0	0	0	0	1	0	PC
0	0	1	0	0	0	0	0	1	1	DR
0	0	0	1	0	0	0	1	0	0	AC
0	0	0	0	1	0	0	1	0	1	IR
0	0	0	0	0	1	0	1	1	0	TR
0	0	0	0	0	0	1	1	1	1	Memory