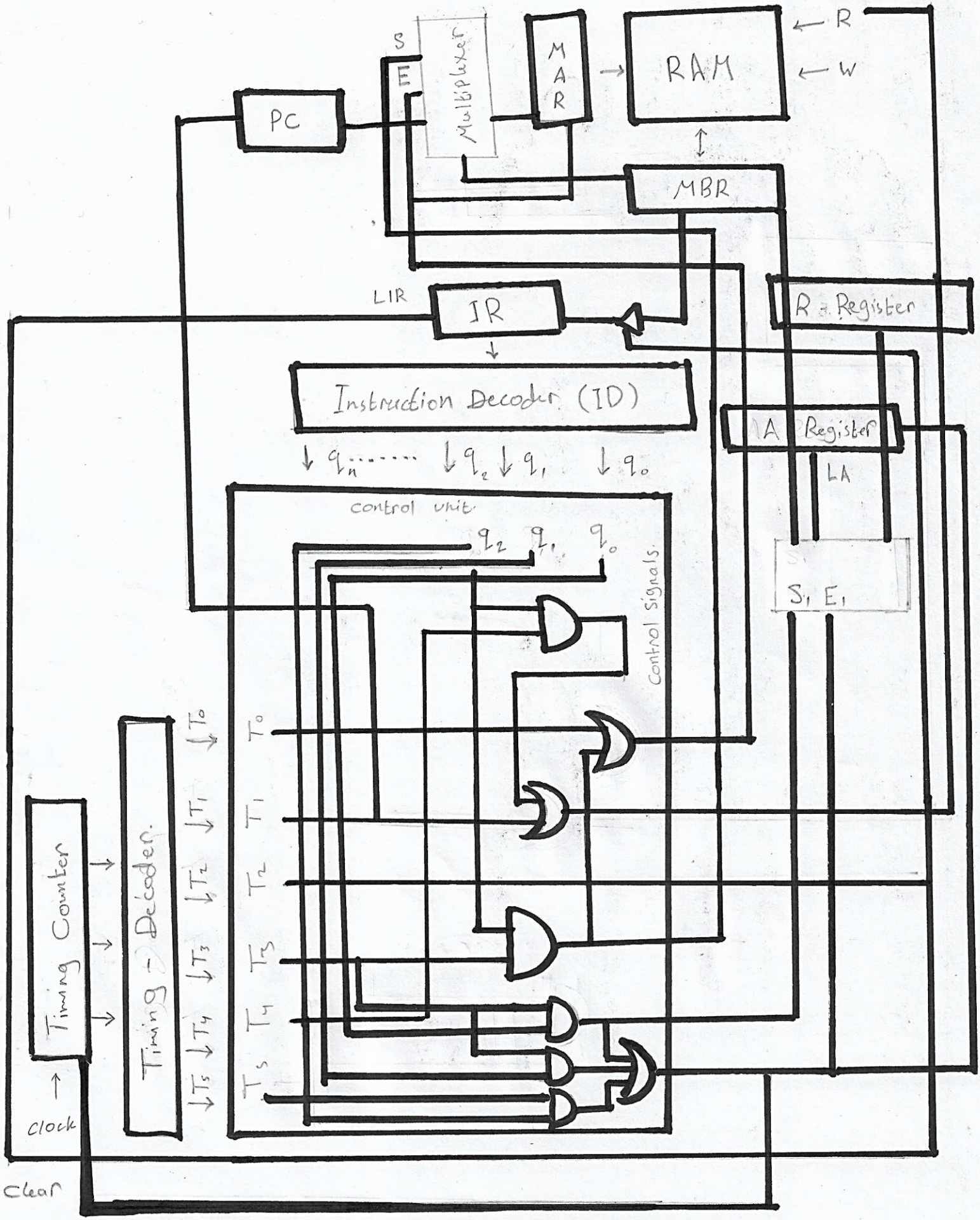

S
E
Multiplexer
MAR
RAM
R
W
PC
MBR
LIR
IR
R = Register
Instruction Decoder (ID)
A Register
LA
control unit
Control Signals
S, E,
Timing Counter
Timing - Decoder
clock
Clear

>> Computer Prototype <<

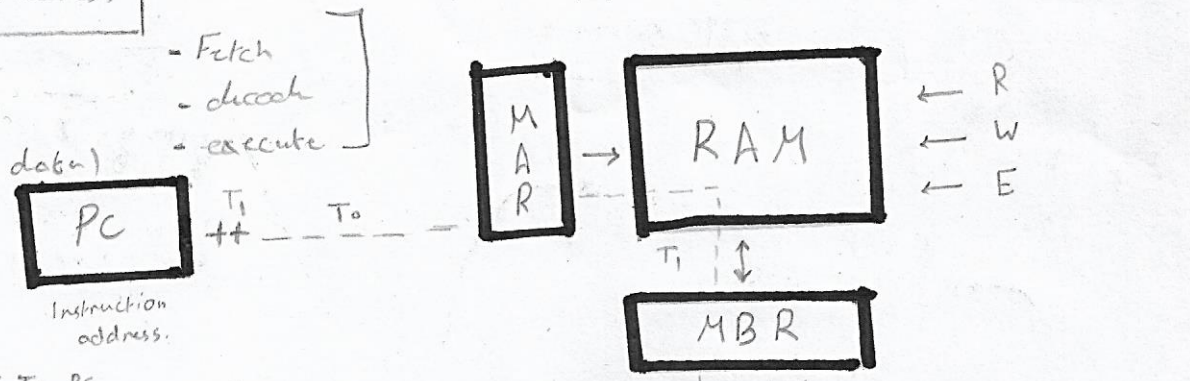
op code	Operand	Address

- Fetch
- decode
- execute

00 A ← R

01 LD Immediate (data)

10 LD Address.



Multiplexer (MAR)

T_0	$T_3 q_2$	L_{MAR}	E_1	S_1
1	x	1	1	0
x	1	1	1	1

[T_0 PC
T_1 PC+1

- $L_{MAR} = T_0 + T_3 q_2$
- $S_1 = T_3 q_2$
- $E_1 = T_0 + T_3 q_2$

op | operand data or address.

op code

IR

T_2

R-Register

Instruction Decoder (ID)

A-Register.

Operation → $q_n \cdots q_2\ q_1\ q_0$

[Memory Enable line]

- $E = T_1 + T_3 q_2$
- $R = T_1 + T_4 q_2$

[Increment PC] - $INC = T_1$

[Transfer to IR] gate (on off)

- $G_1 = T_2$
- $L_2 = T_2$

Control Unit.

Control Signals

→ read
→ clear
→ increment

Multiplexer (A)

$T_3 q_0$	$T_3 q_1$	$T_5 q_2$	L_A	E_2	S_2
1	x	x	1	1	1
x	1	x	1	1	0
x	x	1	1	1	0

- $L_A = T_3 q_0 + T_3 q_1 + T_5 q_2$
- $E_2 = T_3 q_0 + T_3 q_1 + T_5 q_2$
- $S_2 = T_3 q_0$

[Memory Enable line]

- $E = T_1 + T_3 q_2$
- $R = T_1 + T_4 q_2$

[Increment PC] - $INC = T_1$

[Transfer to IR]

- $G_1 = T_2$ (gate on off)
- $L_2 = T_2$

Timing Counter

clear

Clock

Timing Decoder.

T_0 T_1 T_2 T_3 T_4 T_5

[Reset (Timing Counter)]

- Reset = $T_3 q_0 + T_3 q_1 + T_5 q_2$

Condition	Operation.
T_0	- MAR ← PC
T_1	- MBR ← M, PC ← PC+1
T_2	- IR ← MBR [op-code], $q_0 \cdots q_n$ ← Decode [op-code]

- $T_3 q_0$: A ← R, T ← 0 ① Fetch
- $T_3 q_1$: T ← 8 ② Decode
  A ← MBR [operand]
- $T_3 q_2$: MAR ← MBR [address]
- $T_4 q_2$: MBR ← M
- $T_5 q_2$: A ← MBR, T ← 0

③ Execute

corresponding active signal of ID	Operation	op-code
q_0	A ← R	0
q_1	A ← operand	1
q_2	A ← [Address]	2