Ch = 2

(Basic Computer Organization & Design)

1). Instruction format.

→ A computer instruction is often divided into two parts.

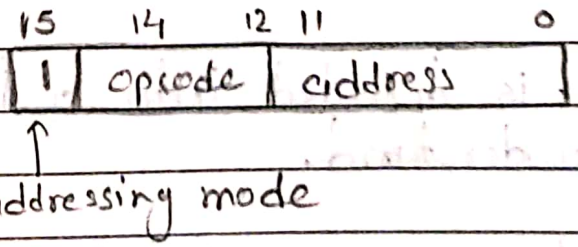
- An opcode (operation code) that specifies the operation for that instruction.
- An address that specifies the resisters and/or locations in memory to use for that operation.

→ In the basic computer, since the memory contains 4096 ($= 2^{12}$) words, we need 12 bit to specify which memory address this instruction will use

→ In the ~~ker~~ basic computer, bit 15 of the instruction specifies the addressing mode (0: direct addressing, 1: Indirect addressing)

→ Since the memory words, and hence the instructions, are 16 bits long, that leaves 3 bits for the ~~ist~~ instruction's opcode.

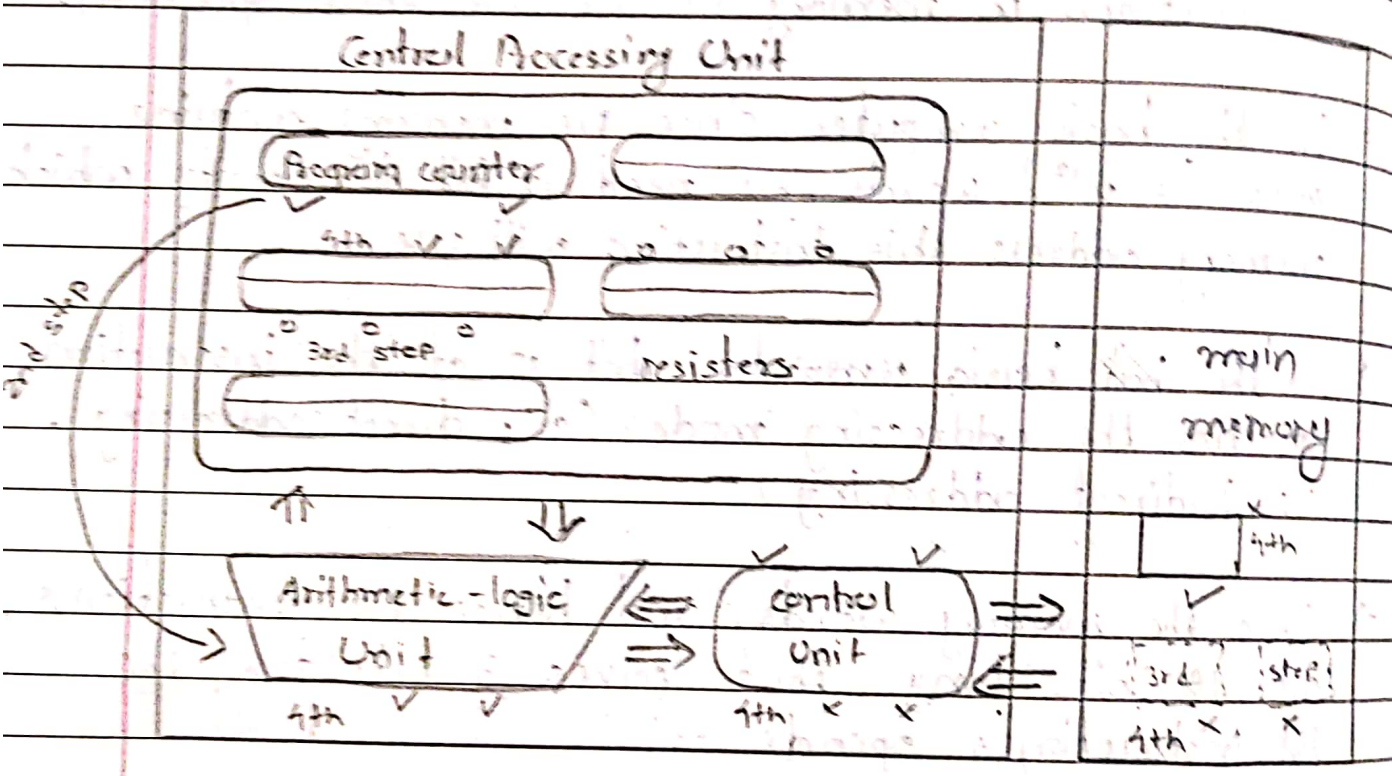
Instruction format.



a) Instruction cycle.

1. Fetch an instruction from memory.

→ The control unit fetches the next instruction from me using the program counter to determine where t instruction is located.



2. Decode the instruction.

→ The instruction is decoded into a language that the ALU can understand.

3. Calculate effective address if indirect address.

→ Any data operands required to execute the instruction are fetched from memory and placed into resisters within the CPU.

Page No.
Date

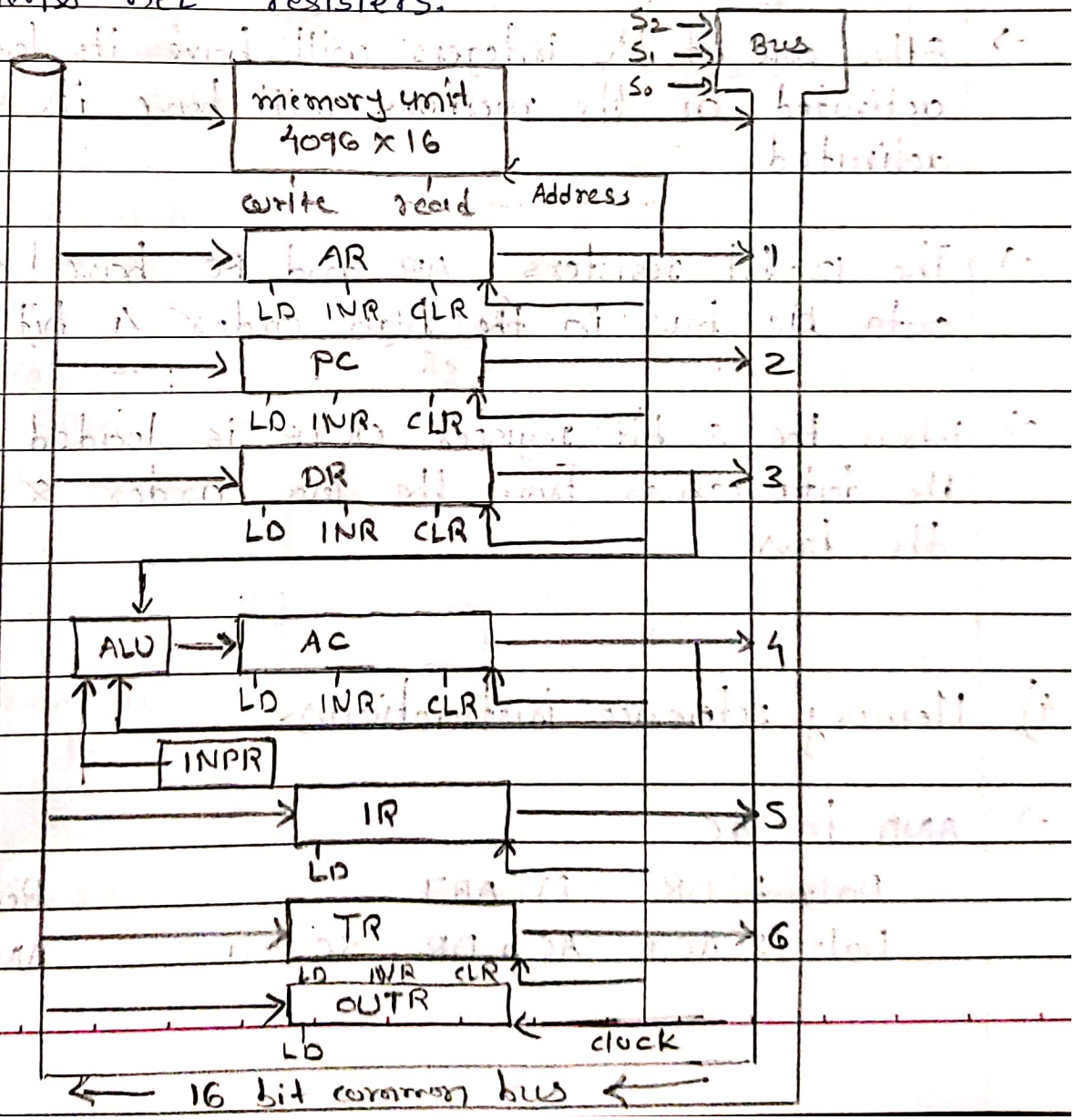
4. Execute instruction.

→ The ALU executes the instruction and places results in registers or memory.

3) Common-bus architecture

→ The registers in the basic computer are connected using a bus.

→ This gives a savings in circuitry over complete connections bet^n registers.



→ Three control lines S_2, S_1 and S_0 control which register the bus selects as its input.

S_2	S_1	S_0	Register
0	0	0	x
0	0	1	AR
0	1	0	PC
0	1	1	DR
1	0	0	AC
1	0	1	IR
1	1	0	TR
1	1	1	Memory

→ Either one of the integers will have its load signal activated, or the memory will have its read signal activated

→ The 12-bit registers, AR and PC have 0's loaded onto the bus in the high order 4 bit positions.

→ When the 8-bit register OUTR is loaded from the bu the data comes from the low order 8-bits on the bus

4), Memory reference instructions.

→ AND to AC

D0T4 : DR ← M[AR] Read operand

D0T5 : AC ← AC ∧ DR, SC ← 0 AND with AC

→ ADD to AC

D_1T_4 : DR ← M[AR] Read operand

D_1T_5 : AC ← AC + DR, E ← Cout, SC ← 0

Add to AC and store carry in E

→ LDA : Load to AC

D_2T_4 : DR ← M[AR]

D_2T_5 : AC ← DR, SC ← 0

→ STA : store AC

D_3T_4 : M[AR] ← AC, SC ← 0

→ BUN : Branch Unconditionally.

D_4T_4 : PC ← AR, SC ← 0

→ BSA : Branch and Save return address

M[AR] ← PC, PC ← AR + 1

Memory PC, AR at time T_4

20	0 BSA 135	
PC = 21	Next Instruction	
AR = 135		
136	Subroutine	
	↓	
	1 BUN 135	

Memory

Memory PC, After execution

20	0 BSA 135	
21	Next Instruction	
135		
PC = 136	Subroutine	
	↓	
	1 BUN 135	

Memory

→ BSA :

D_5T_4 : $M[AR] \leftarrow PC$, $AR \leftarrow AR + 1$

D_5T_5 : $PC \leftarrow AR$, $SC \leftarrow 0$

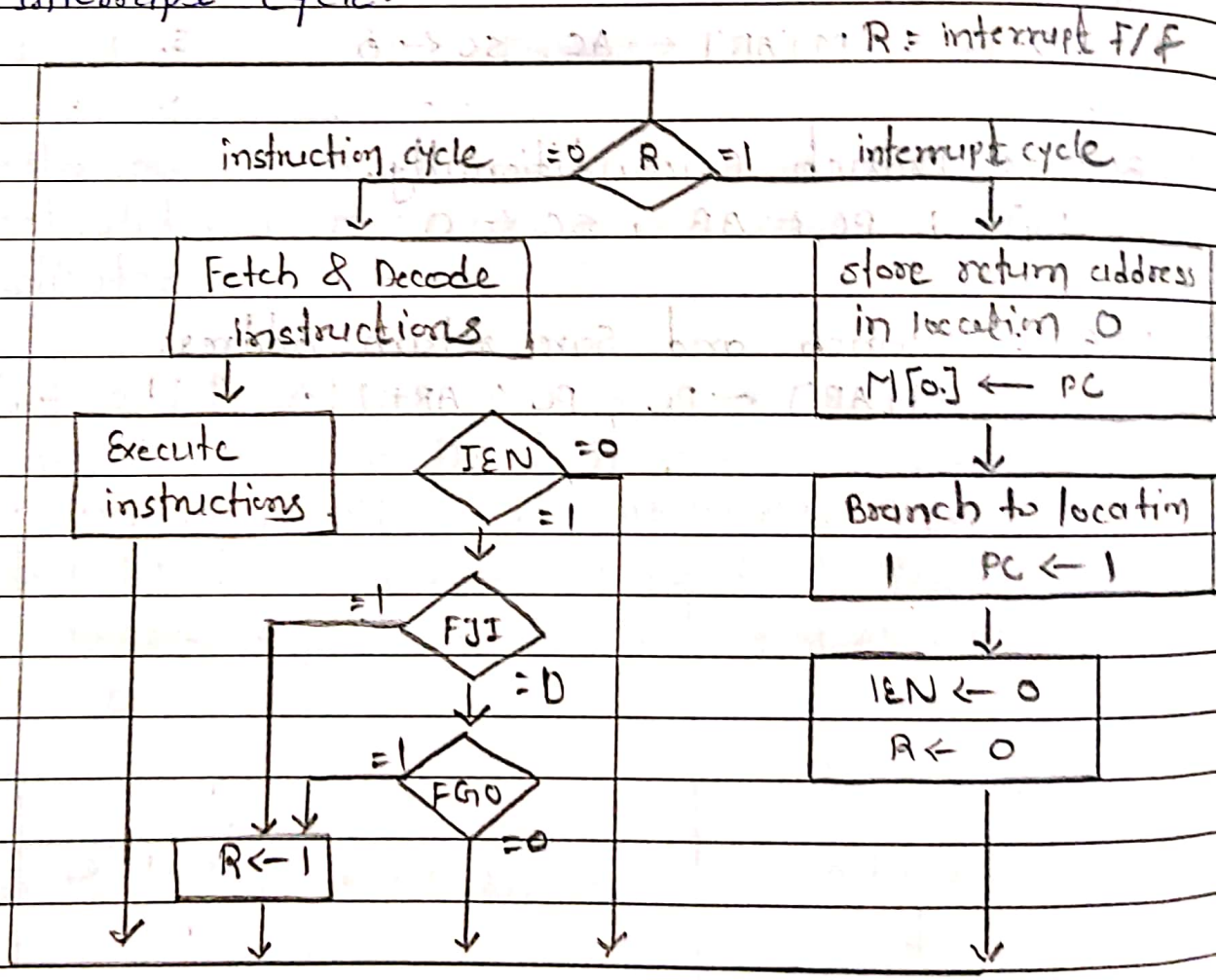
→ ISZ : Increment and Skip-if-zero

D_6T_4 : $DR \leftarrow M[AR]$

D_6T_5 : $DR \leftarrow DR + 1$

D_6T_6 : $M[AR] \leftarrow DR$, if $(DR = 0)$ then $(PC \leftarrow PC + 1)$, $SC \leftarrow$

5). Interrupt cycle.



→ The interrupt cycle is a HW implementation of a bra
and save return address operation.

→ At the begining of the next instruction cycle, the instruction that is read from memory is in address 1.

→ At memory address 1, the programmer must store a branch instruction that sends the control to an interrupt service routine.

Memory

Before interrupt

0		
1	0 BUN 1120	
255	Main	
256	Program	
1120	I/O	
	Program	
	1 BUN 0	

After interrupt

	0	256
PC = 1	0 BUN 1120	
	255	Main
	256	Program
	1120	I/O
		program
	1 BUN 0	

Register transfer statements for interrupt cycle.

- if IEN (FGI + FGO) $T_0' T_1' T_2'$ ↔

$T_0' T_1' T_2'$ (IEN) (FGI + FGO) : R ← 1

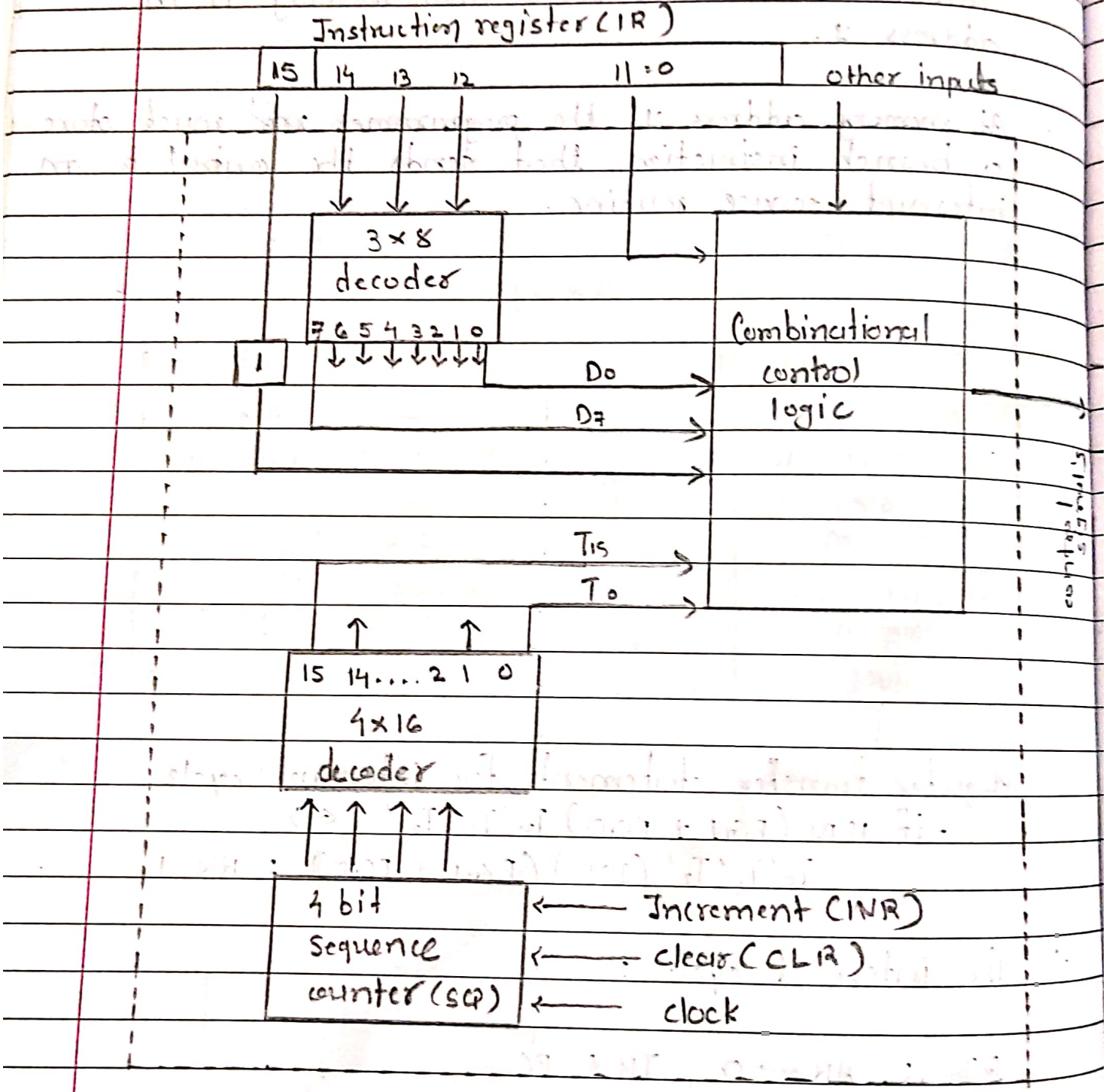
→ The interrupt cycle :

RT_0 : AR ← 0, TR ← PC

RT_1 : M[AR] ← TR, PC ← 0

RT_2 : PC ← PC + 1, IEN ← 0, R ← 0, SC ← 0

8). Design of control unit of basic computer.



-> The 8 outputs of decoder are designed by the symbols D0 to D7. Bit 15 of the instruction is transferred to a flip flop designed by the symbol I. Bits 0

through 15 are applied to the control logic gates. The 4-bit sequence counter can count in binary from 0 through 15.

→ The o/p of the counter are decoded into 16 timing signals T_0 through T_{15}.

6), Difference : Direct and Indirect addressing modes.

Direct addressing Mode	Indirect addressing Mode
1. Address field contains the effective address of operand.	1. Address field contains the refference of effective address.
2. Requires only one memory reference.	2. Requires two memory references.
3. Fast addressing	3. Slower than direct addressing mode.
4. No further classification	4. Further classified into two categories.
5. No further calculation is required to perform the operation.	5. Require further calculatio to find the effective address.

7). Difference : Hardwired & micro program control unit.

No.	Attributes	Hardwired control Unit	Microprogrammed control unit.
1.	Speed	Speed is fast	Speed is slow
2.	Cost of implementation	More costlier	Cheaper
3.	Flexibility	Not flexible to accommodate new system specification or new instruction redesign is required	More flexible to accommodate new system specification or new instruction sets.
4.	Ability to handle complex instructions	Difficult to handle complex instruction sets.	Easier to handle complex instruction sets.
5.	Decoding	Complex decoding and sequencing logic	Easier decoding and sequencing logic
6.	Applications	RISC Microprocessor	CISC Microprocessor
7.	Instruction set of size	Small	Large.

8.	Control memory	Absent	Present.
9.	Chip area required	Less	More
10.	Occurrence	Occurrence of error is more	Occurrence of error is less.