INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE: CENTRATOM TRIESTE

H4.SMR. 403/10

FIFTH COLLEGE ON MICROPROCESSORS: TECHNOLOGY AND APPLICATIONS IN PHYSICS

2- 27 October 1989

Fundamentals of Digital Electronics: Design Style

S. VENKATARAMAN
Microprocessors Laboratory, Trieste, Italy

DESIGN STYLE

GUIDELINES FOR GOOD DESIGN STYLE

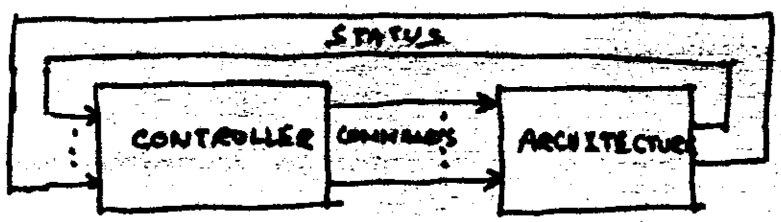
(i) DESIGN FROM TOP DOWN.

(ii) DISTINGUISH BETWEEN THE CONTROLLER & CONTROLLED HARDWARE.

(iii) DEVELOP & CLEARLY DEFINE THE ARCHITECTURE & CONTROL ALGORITHM BEFORE DECIDING THE HARDWARE.

— * —

STRUCTURE OF A STATE MACHINE



- PARTITION DESIGN INTO CONTROL ALGORITHM & ARCHITECTURE
- CONTROL ALGORITHM IS USUALLY INDEPENDENT OF HARDWARE.
- THE ARCHITECTURE INCLUDE A DETAILED SET OF COMPONENTS DATA PATHS FOR CONTROLLED DEVICE, THE CONTROL SIGNALS THEY REQUIRE & THE STATUS OUTPUTS THEY PRODUCE.
- THE CONTROLLER PRODUCES PROPERLY SEQUENCED COMMANDS TO MAKE THE ARCHITECTURE EXECUTE YOUR PROBLEM.

DESIGN YOUR CONTROLLER FIRST AND YOU WOULD HAVE DESIGNED THE HARDWARE!

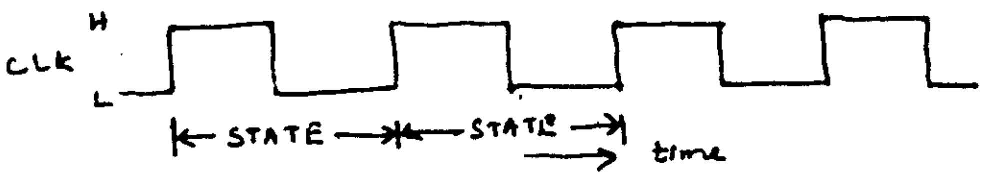
- WE DESCRIBE SYNCHRONOUS MACHINES.
- WE DEVELOP AN ASM CHART TO DESCRIBE THE MACHINE'S CONTROLLER.

STATES

- AN ASM MOVES THRU' A SEQUENCE OF **STATES**, BASED ON THE POSITION OF THE CONTROL ALGORITHM AND THE VALUE OF RELEVANT STATUSES.

- A **STATE** DESCRIBES SUFFICIENT KNOWLEDGE OF THE PRESENT & PAST CONDITIONS TO DETERMINE THE FUTURE.

- THE STATE TIMES ARE DETERMINED BY A MASTER CLOCK, USUALLY A SQUARE WAVE VOLTAGE SIGNAL

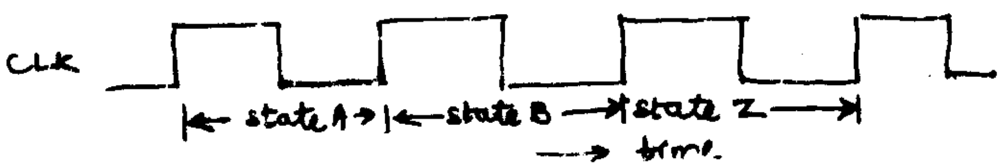


ASM CHART NOTATIONS

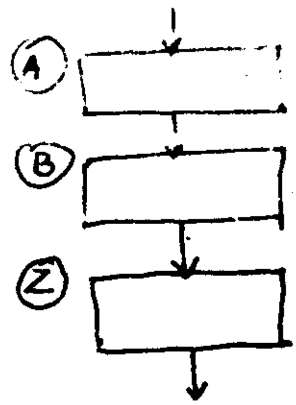
STATES:

- EACH ACTIVE CLOCK TRANSITION CHANGE OF STATE FROM THE PRESENT TO THE NEXT.

FOR EXAMPLE:



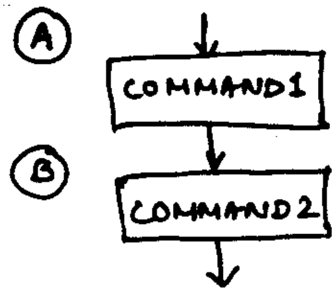
THE ASM CHART WILL BE



OUTPUTS

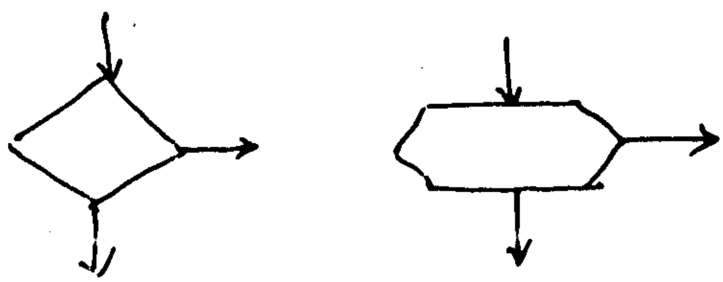
- THE CONTROLLER SENDS WELL SEQUENCED OUTPUTS TO THE CONTROLLED DEVICE ACCORDING TO SOME ALGORITHM

FOR EXAMPLE: IF IN STATE A THE OUTPUT IS COMMAND 1 & IN STATE B IF THE OUTPUT IS COMMAND2, THE ASM CHART WILL LOOK LIKE THIS

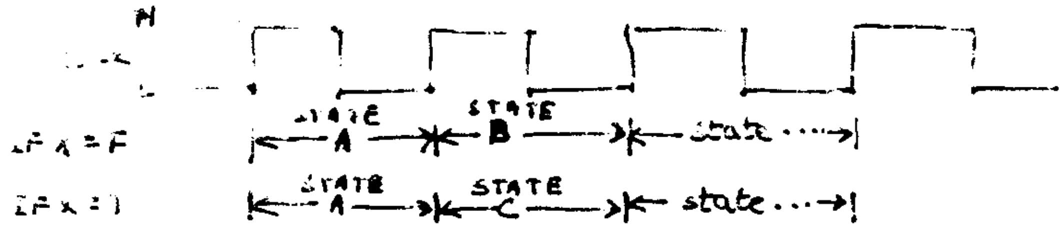


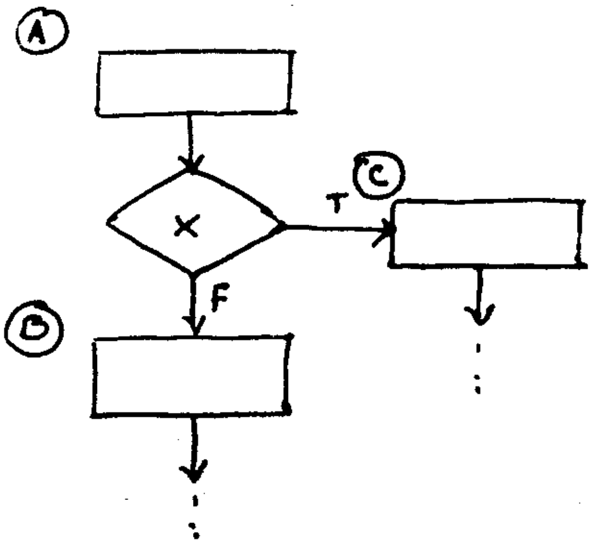
BRANCHES

- PURELY SEQUENTIAL ASMs ARE NOT POWERFUL ENOUGH. TO DESCRIBE ALGORITHMS YOU WOULD LIKE TO.
- WE WOULD LIKE TO DESCRIBE **CONDITIONAL BRANCHES** SO THAT THE NEXT STATE IS NOT ONLY DETERMINED BY THE PRESENT STATE VALUE BUT ALSO BY THE PRESENT VALUE OF ONE OR MORE <u>STATUS</u> INPUTS.



FOR EXAMPLE:

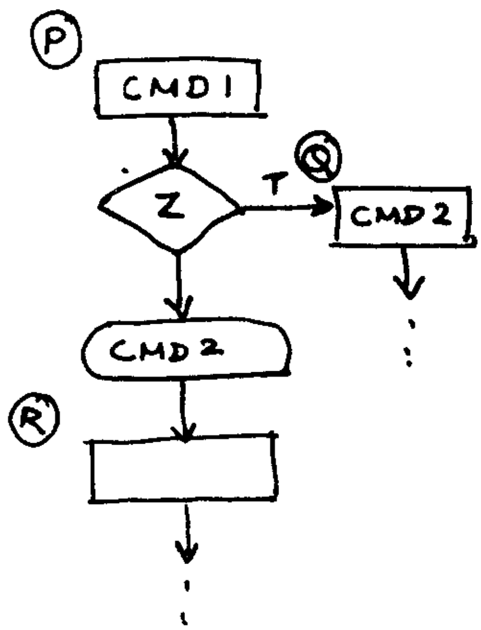




CONDITIONAL OUTPUTS

- WE MAY SOMETIMES REQUIRE THAT THE CONTROLLER PRODUCE AN OUTPUT WHILE IN SOME STATE PROVIDED SOME STATUS INPUT CONDITIONS ARE MET.

FOR EXAMPLE:



- THE TWO STATE VARIABLES B & A SPECIFY AN ADDRESS THAT POINTS TO THE PRESENT STATE.
- IF WE COMPUTE THE NEXT ADDRESS AND PUT INTO THE FLIPFLOPS, WE WOULD BE POINTING TO A NEW STATE.
- WE CAN USE COMBINATIONAL CIRCUITS TO COMPUTE THE NEXT ADDRESS.
- WE USE D-FLIP FLOPS FOR THE DESIGN.
- THE ASM CHART OF OUR EXAMPLE HAS ONE STATUS INPUT Z, TWO COMMAND OUTPUTS, CMD1 & CMD2 & TWO STATE FLIPFLOPS B & A.
- THE COMBINATIONAL LOGIC MUST FOLLOW THE FOL TABLE SHOWN BELOW.

PRESENT			NEXT	
B	A	Z	B(D)	A(D)
0	0	0	1	0
0	0	1	1	1
0	1	x	0	0
1	0	x	0	0
1	1	x	0	0

- NOTICE THAT WE HAVE NOT ASSIGNED THE STATE 01 IN THE ASM CHART BUT HAVE INCLUDED IN THE TABLE.
- IF WE ENTER INTO THE UNASSIGNED STATE 01 DURING POWER ON WE GET OUT OF THE STATE TO STATE 00 IN THE NEXT CLOCK CYCLE.

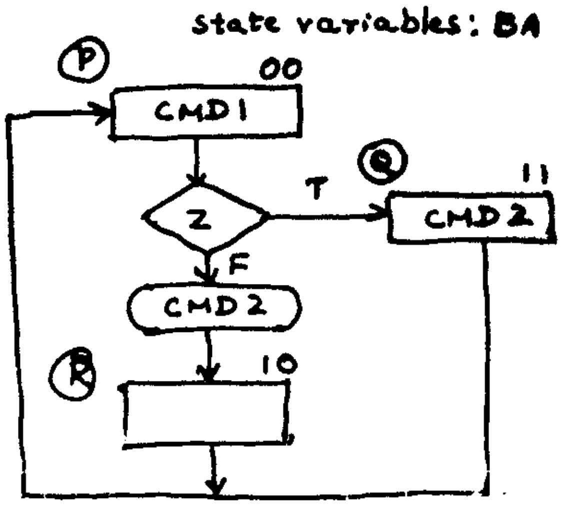
REALIZING ASM

- TASK IS TO CONSTRUCT A STATE GENERATOR FOR A GIVEN ASM.
- THE STATE GENERATOR RECORDS THE PRESENT STATE AND GENERATES THE NEXT STATE
- TO KEEP TRACK OF THE PRESENT STATE WE USE FLIP-FLOPS.
- WE CAN ENCODE EACH STATE AS A BINARY NUMBER. THEREFORE n FLIP-FLOPS WILL BE REQUIRED TO GENERATE 2^n STATES.
- WE CAN ALSO AVOID ENCODING BY ASSIGNING A FLIP FLOP FOR EACH STATE.

TRADITIONAL SYNTHESIS

- USES AN ENCODED REPRESENTATION OF THE PRESENT STATE TO COMPUTE THE NEXT STATE.
- THE BITS OF THE CODE ARE THE STATE VARIABLES. n STATE VARIABLES DESCRIBE 2^n STATES.
- WE ASSIGN ARBITRARILY STATES OF BINARY STATE VARIABLES.

EXAMPLE:



MULTIPLEXER METHOD OF SYNTHESIS

THE TRADITIONAL METHOD HAS THE FOLLOWING DISADVANTAGES:

- IT HAS NO ONE TO ONE CORRESPONDENCE BETWEEN THE ASM CHART & THE HARDWARE IT REPRESENTS.
- EVERY CHANGE IN THE ALGORITHM HOWEVER ~~SMALL,~~ REQUIRES A FRESH DESIGN OF THE NEXT STATE COMBINATIONAL LOGIC.

THE MULTIPLEXER METHOD HAS THE FOLLOWING ATTRIBUTES:

- IT PRODUCES A DESIGN THAT HAS A DIRECT CORRESPONDENCE WITH THE ALGORITHM.
- IT CAN BE APPLIED TO ANY ASM CHART.

IN THIS DESIGN METHOD

- STATE ASSIGNMENTS ARE DONE AS IN THE PREVIOUS METHOD.
- THE APPROPRIATE NUMBER OF D-FLIP FLOPS FOR THE STATE VARIABLES ARE CHOSEN.
- TO PRODUCE THE NEXT STATE CODE, WE USE MULTIPLERS INSTEAD OF COMBINATIONAL LOGIC.
- A MULIPLEXER IS USED FOR THE INPUT OF EACH STATE FLIP FLOP.
- THE MULTIPLEXER HAS THE SAME NUMBER OF INPUTS AS THE NUMBER OF STATES.
- A STATE TRANSITION TABLE WRITTEN BY LOOKING AT THE ASM CHART DECIDES THE INPUT DATA FOR THE MULTIPLEXER.

- THE EQUATIONS FOR THE STATE FLIP-FLOP INPUTS B(D) AND A(D) ARE (FROM THE TABLE)

$$B(D) = \overline{B}.\overline{A}.\overline{Z} + \overline{B}.\overline{A}.Z = \overline{B}.\overline{A}$$

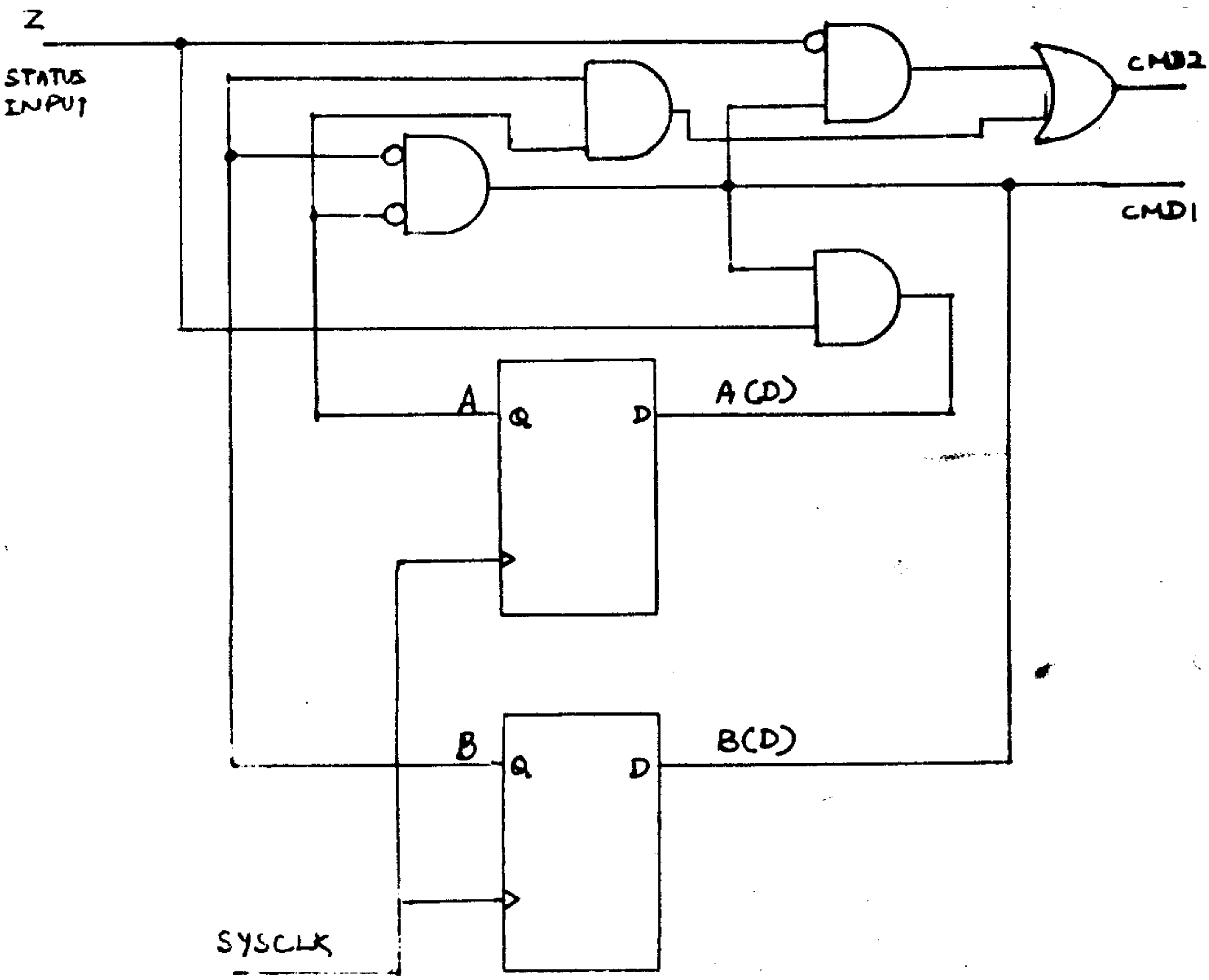
$$A(D) = \overline{B}.\overline{A}.Z$$

- THE EQUATIONS FOR THE OUTPUTS ARE

$$CMD1 = STATE\ P = \overline{B}.\overline{A}$$

$$CMD2 = STATE\ P.\overline{Z} + STATE\ Q = \overline{B}.\overline{A}.\overline{Z} + B.$$

- THE HARDWARE IS AS SHOWN BELOW.



STATE FLIPFLOPS

EXAMPLE [illegible]: LET US TAKE THE PREVIOUS EXAMPLE'S ASM CHART.

- WE HAVE THE TRANSITION TABLE AS FOLLOWS:

PRESENT STATE NUMBER	PRESENT STATE NAME	NEXT STATE NAME	B	A	CONDITION FOR TRANSITION
0	P	R	1	0	$\bar{Z}$
		Q	1	1	Z
2	R	P	0	0	T
3	Q	P	0	0	T
1	-	P	0	0	T

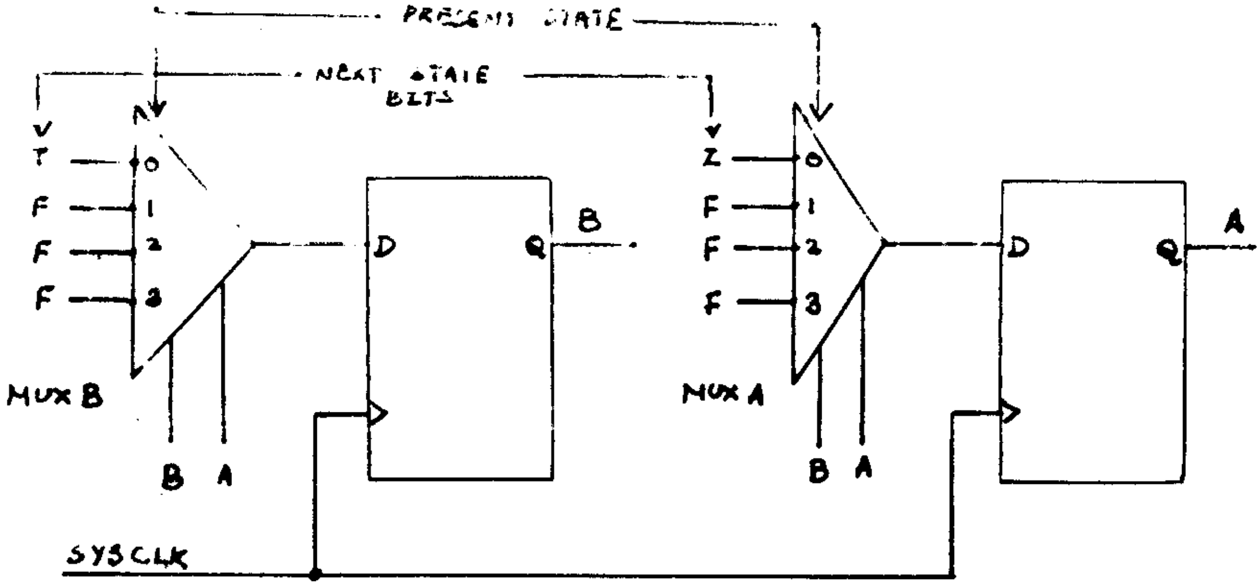
- SINCE WE HAVE THREE STATES AS DEFINED BY THE ASM CHART A 4-INPUT MUX FOR EACH FLIP-FLOP IS REQUIRED.
- TWO D-FLIP-FLOPS ARE REQUIRED TO STORE THE PRESENT STATE.
- WE CALL THE MUXES THAT INPUT TO EACH OF THE D-FLIPFLOPS AS MUXA & MUXB.
- FROM THE TRANSITION TABLE WE CAN WRITE THE EXPRESSIONS FOR MUXB INPUTS AS:

$$MUXB(0) = \bar{Z} + Z = T.$$
$$MUXB(1) = F$$
$$MUXB(2) = F$$
$$MUXB(3) = F$$

- SIMILARLY THE EXPRESSIONS FOR MUXA INPUTS ARE:

$$MUXA(0) = Z$$
$$MUXA(1) = F$$
$$MUXA(2) = F$$
$$MUXA(3) = F$$

THE [illegible] OF THE ASM [illegible] AS FOLLOWS.



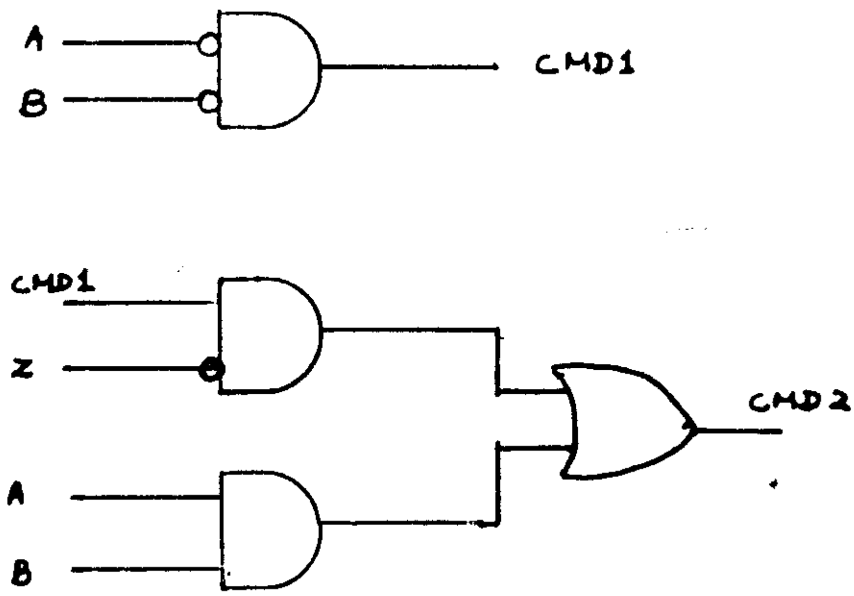
THE EQUATION FOR THE OUTPUTS ARE:

$$CMD1 = STATE\ P = \overline{B} \cdot \overline{A}$$

$$CMD2 = STATE\ P \cdot \overline{Z} + STATE\ Q = \overline{B} \cdot \overline{A} \cdot \overline{Z} + B \cdot A$$

$$= CMD1 \cdot \overline{Z} + B \cdot A$$

THE OUTPUTS CAN BE SYNTHESISED DIRECTLY FROM THE EQUATIONS

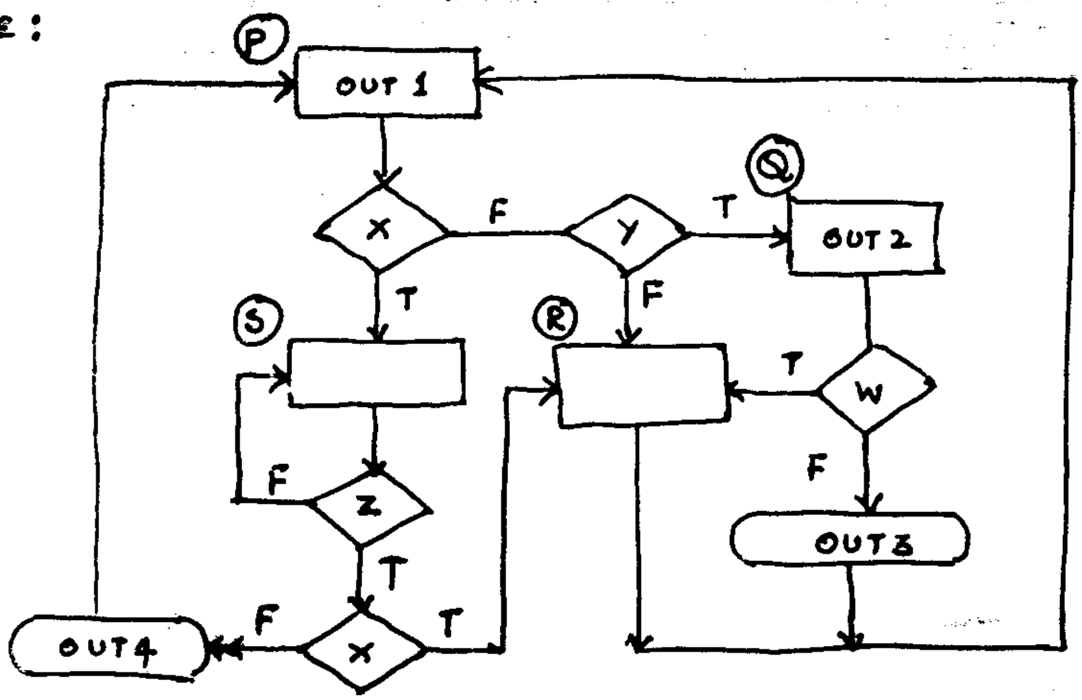


THE ONE-HOT METHOD OF SYNTHESIS

IN THIS METHOD OF GENERATING STATES

- THERE IS NO ENCODING OF STATES & WE USE ONE FLIP-FLOP FOR EACH STATE.
- SINCE WE HAVE TO BE IN ONE STATE AT A TIME, ONLY ONE OF THE STATE-FLIPFLOPS MUST BE TRUE DURING EACH STATE TIME.
- WE COMPUTE WITH COMBINATIONAL LOGIC THE INPUTS OF THE STATE-FLIPFLOPS.

EXAMPLE:



- FOR THIS ASM SINCE THERE ARE 4-STATES WE WOULD REQUIRE 4-D FLIP-FLOPS
- THE INPUTS TO THESE FLIP-FLOPS CAN BE COMPUTED THRU' COMBINATIONAL LOGIC, MADE BY ANALYZING THE STATE TRANSITION TABLE.

THE STATE TRANSITION TABLE FOR THE ASM IS AS SHOWN BELOW:

NEXT STATE	PRESENT STATE	CONDITION FOR TRANSITION
P	Q	$\overline{W}$
	R	T
	S	$Z.\overline{X}$
Q	P	$\overline{X}.Y$
R	P	$\overline{X}.\overline{Y}$
	Q	W
	S	$X.Z$
S	P	X
	S	$\overline{Z}$

THE EQUATIONS FOR THE FLIPFLOP INPUTS ARE:

NEXT STATE.P = $P(D) = Q.\overline{W} + R + S.Z.\overline{X}$

NEXT STATE.Q = $Q(D) = P.\overline{X}.Y$

NEXT STATE.R = $R(D) = P.\overline{X}.\overline{Y} + Q.W + S.X.Z$

NEXT STATE.S = $S(D) = P.X + S.\overline{Z}$

THE EQUATIONS FOR THE OUTPUTS ARE:

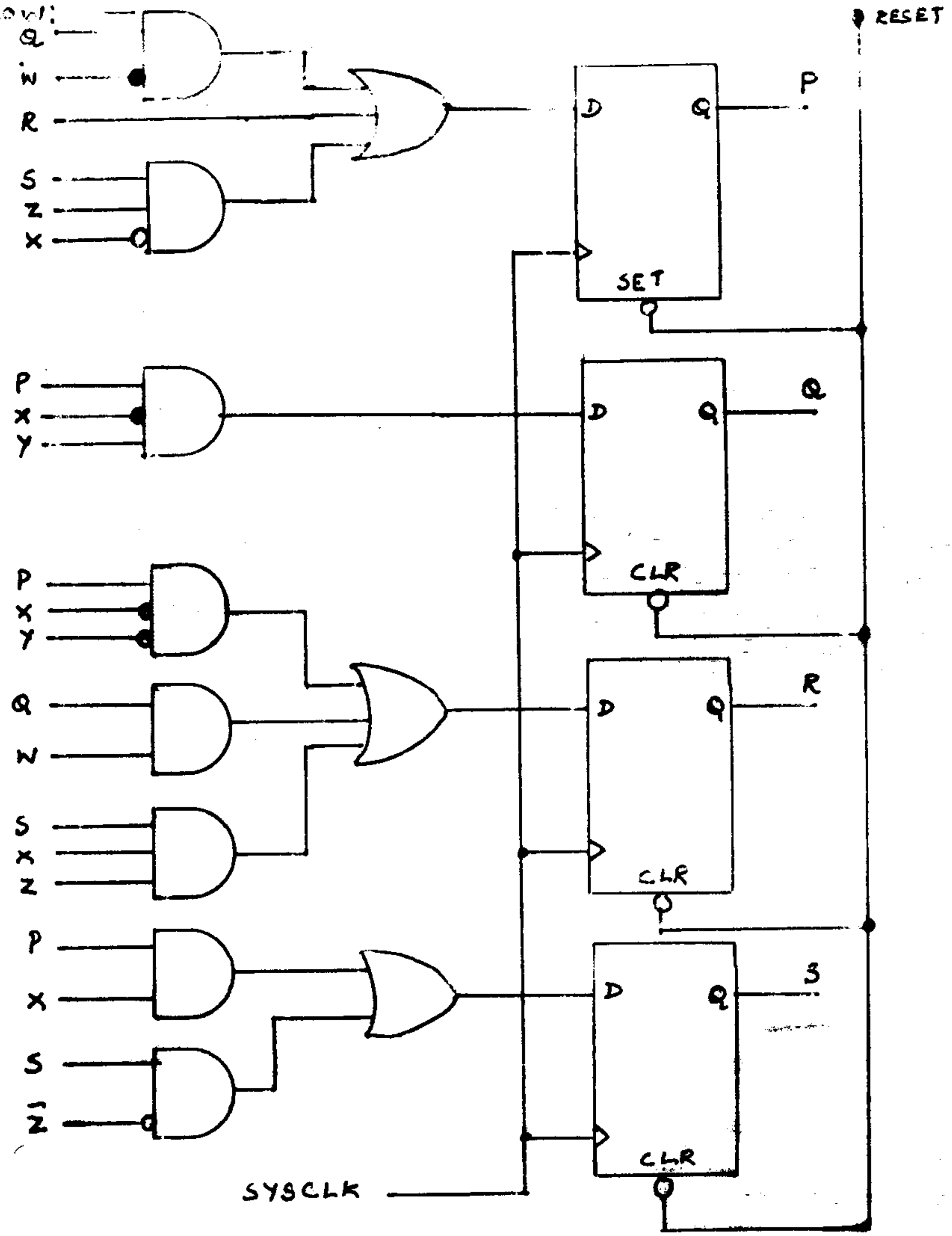
OUT 1 = P.

OUT 2 = Q

OUT 3 = $Q.\overline{W}$

OUT 4 = $S.Z.\overline{X}$

THE IMPLEMENTATION OF THE CONTROLLER IS AS SHOWN BELOW:



INITIALIZATION OF THE ONE-HOT CONTROLLER

- IN THE EARLIER DESIGN METHODS WHICH REQUIRED AN ENCODED REPRESENTATION OF THE STATE, THE CODE INHERENTLY SPECIFIES ONLY ONE STATE AT A TIME.
- IN THE ONE-HOT DESIGN METHOD, WE WILL HAVE TO INITIALIZE THE SYSTEM IN SUCH A WAY THAT ONLY ONE FLIP-FLOP - REPRESENTING THE STARTING STATE IS TRUE AND THE REST FALSE.

THE ROM-BASED METHOD OF SYNTHESIS

- THIS METHOD IS BASED ON TABLE-LOOK UP FROM A PERMANENT MEMORY SUCH AS A ROM.
- THIS METOD IS VERY REGULAR IN APPROACH
- IT HAS THE DIADVANTAGE THAT THE SIZE OF THE ROM INCREASES AS ASMs BECOME COMPLEX.

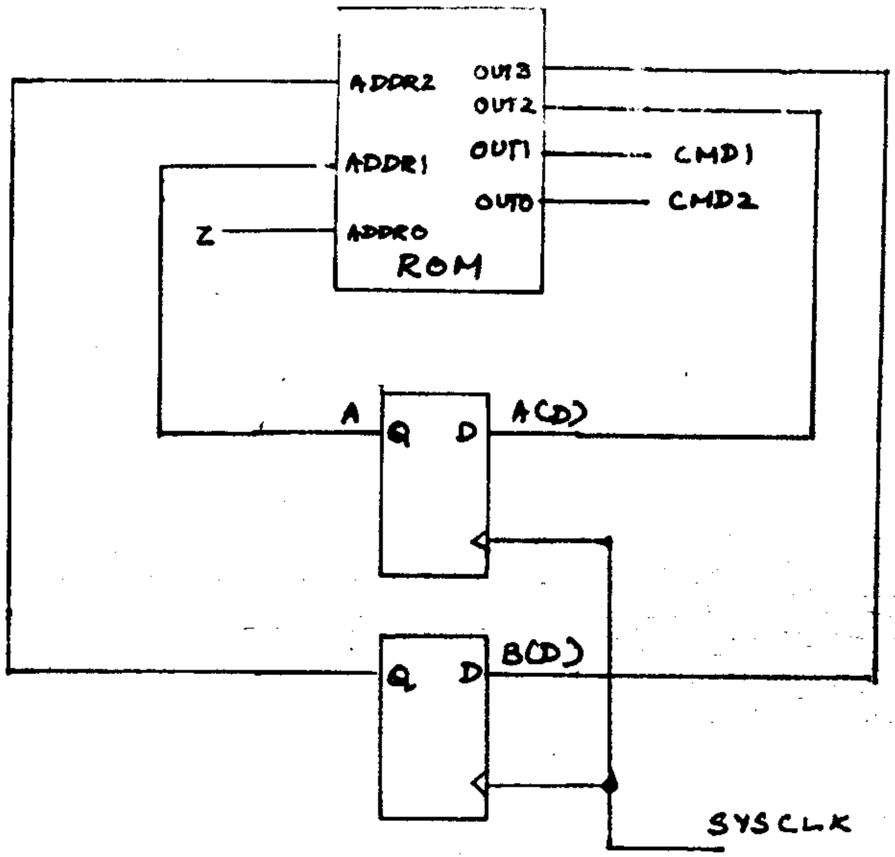
EXAMPLE: WE USE THE FIRST ASM CHART FOR THIS EXAMPLE

- WE USE A ROM TO GENERATE THE NEXT STATE OF THE D-FLIP FLOPS.
- WE USE THE ENCODED REPRESENTATION FOR THE STATES.
- TO IMPLEMENT THE ASM WE CHOOSE A SUITABLE ROM AND SPECIFY ITS CONTENTS.

THE ROM CONTENTS CAN BE DRAWN AS A TABLE:

ADDRESS			OUTPUT			
B	A	Z	B(D)	A(D)	CMD1	CMD2
0	0	0	1	0	1	1
0	0	1	1	1	1	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	1
1	1	1	0	0	0	1

THE FOLLOWING DIAGRAM IS THE IMPLEMENTATION:



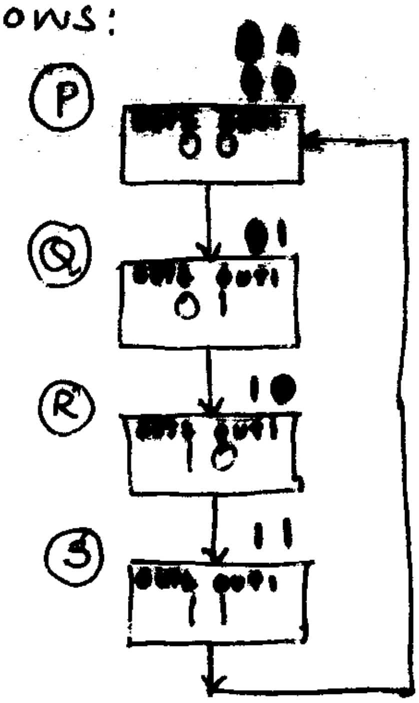
- AS IS SEEN IN THIS IMPLEMENTATION, THE ADDRESS LINES FOR THE ROM INCREASES LOGRITHMICALLY WITH THE NUMBER OF ASM STATES.
- BUT THE ROM HAS AN ADDRESS FOR EACH TEST INPUT & THEREFORE THE ROM SIZE GROWS EXPONENTIALLY AS THE NUMBER OF INPUTS.

DESIGN EXAMPLE

DESIGN OF A MOD-4 COUNTER.

WHAT DOES A MOD-4 COUNTER DO?
WITH EACH SYSTEM CLOCK CYCLE THE MOD-4 COUNTER CYCLICALLY STEPS THROUGH 4-STATES, OUTPUTTING THE VALUE OF THE STATE IT IS IN.

THE ASM CHART FOR THE MOD-4 COUNTER IS AS FOLLOWS:



AS THE NEXT STEP WE CHOOSE TO ENCODE THE STATE, SUCH THAT IT IS EQUAL TO THE VALUE BEING OUTPUT IN EACH STATE.

THIS WE DO, BECAUSE IT EASES THE DESIGN. AND WE NEED NOT BUILD A DECODER TO DECODE THE OUTPUT FROM THE STATE FLIP FLOPS. THE OUTPUT OF THE STATE FLIP FLOPS THEMSELVES ARE THE OUTPUTS.

- THERE WILL BE TWO STATE FLIP FLOPS TO STORE THE PRESENT STATE (B & A).

- THE STATE TRANSITION DATA CAN BE WRITTEN BY LOOKING AT THE ASM CHART.

PRESENT STATE			NEXT STATE		
NAME	B	A	NAME	B(D)	A(D)
P	0	0	Q	0	1
Q	0	1	R	1	0
R	1	0	S	1	1
S	1	1	P	0	0

- THE EQUATIONS FOR THE STATE-FLIPFLOP INPUTS CAN BE WRITTEN FROM THE STATE TRANSITION DATA.

$$A(D) = \bar{B}.\bar{A} + B.\bar{A} = (\bar{B}+B).\bar{A} = \bar{A}.$$

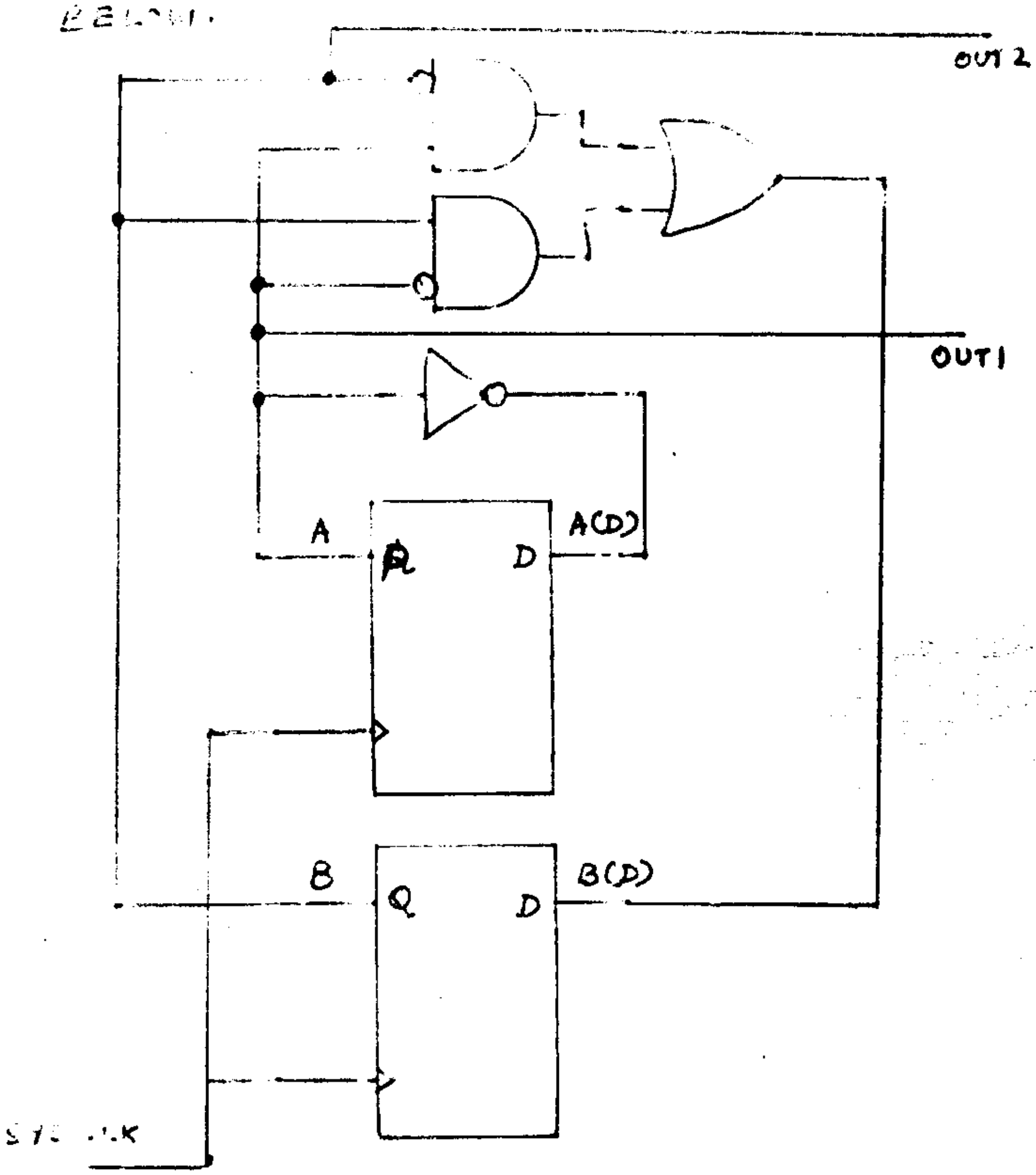
$$B(D) = \bar{B}.A + B.\bar{A}$$

- THE OUTPUTS OF THE MOD-4 COUNTER WILL THEN BE (FROM THE ASM CHART).

$$OUT1 = STATE\,Q + STATE\,S$$
$$= \bar{B}.A + B.A = (\bar{B}+B).A = A$$

$$OUT2 = STATE\,R + STATE\,S$$
$$= B.\bar{A} + B.A = B.(\bar{A}+A) = B$$

THE HARDWARE OF THE MOD-4 COUNTER IS SHOWN BELOW.



EXCERCISE DESIGN A MOD-5 COUNTER.