

# Chapter 8:

8.1 → 8.4

8-2

## Multiplication.

$$\begin{array}{r} 1011 \\ \times 101 \\ \hline 1011 \leftarrow \text{copy} \\ 0000 \leftarrow \text{shift} \\ 1011 \leftarrow \text{copy \& shift} \\ \hline 110111 \leftarrow \text{Add all of them.} \end{array}$$

Again:

$$\begin{array}{r} 1011 \\ \times 101 \\ \hline 0000 \leftarrow \text{all 0s initially} \\ 1011 \\ \hline 1011 \leftarrow \text{1st addition} \\ 01011 \leftarrow \text{shift for 1st addition} \\ 001011 \leftarrow \text{shift for the 0.} \\ 1011 \leftarrow \text{copy \& Add} \\ \hline 110111 \\ \leftarrow \text{final shift} \\ 0110111 \end{array}$$

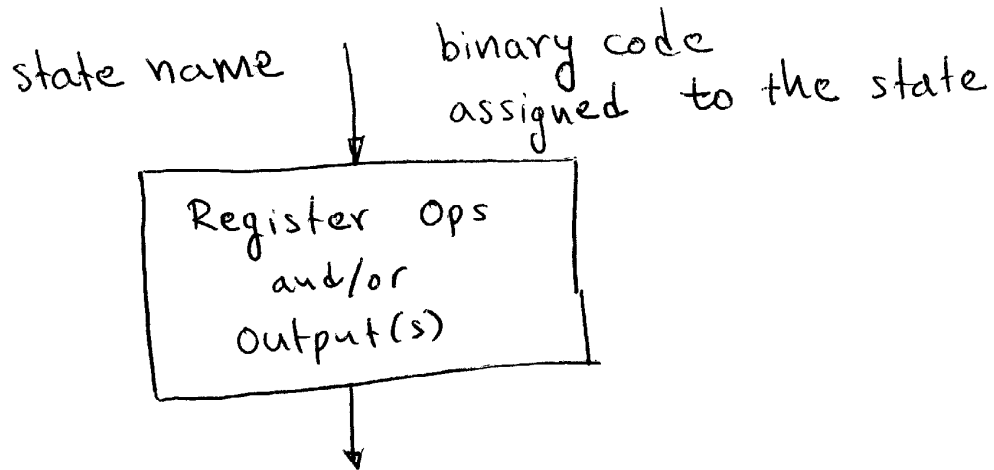
# 8-2 Algorithmic State Machines

8-2-1

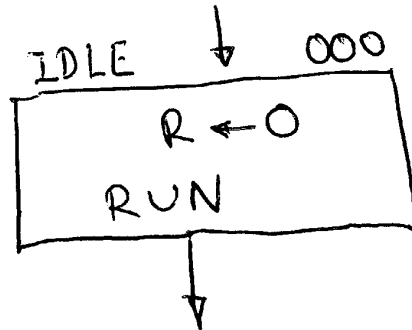
state machine = sequential circuit

A SM chart = state transition diagram + timing.

Ex: state box = node in transition diagram



state eg:



means:

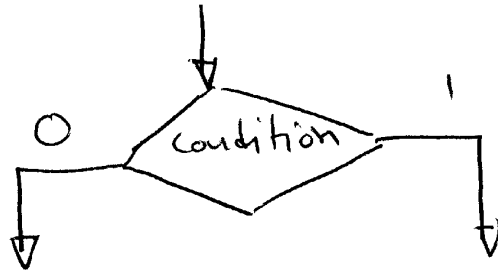
- (i) state = IDLE
- (ii) states are represented by 3 FFs

(iii) IDLE = 000

(iv) In IDLE:

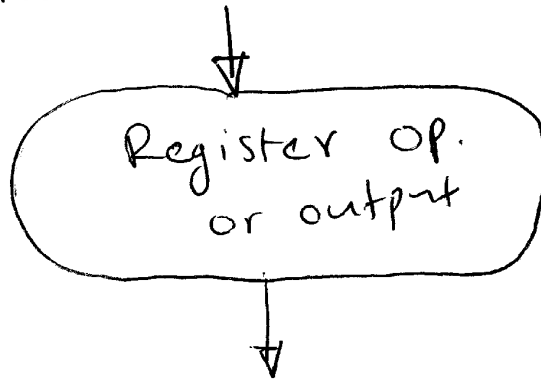
→ R ← 0  
→ output RUN = 1

Decision box: based on condition = input.



Conditional output box:

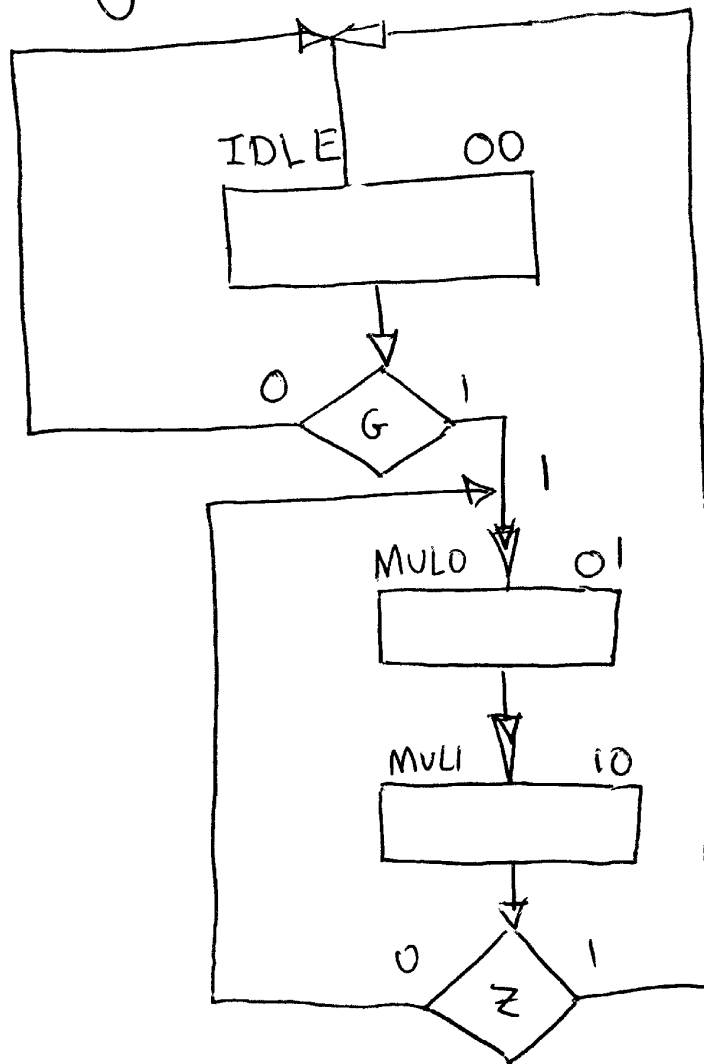
From decision box



Note:

(i) Register transfers only occur when a clock event occurs.

Consider deriving the state transition diagram, and state transition table from Figure 8-8:



Observe:

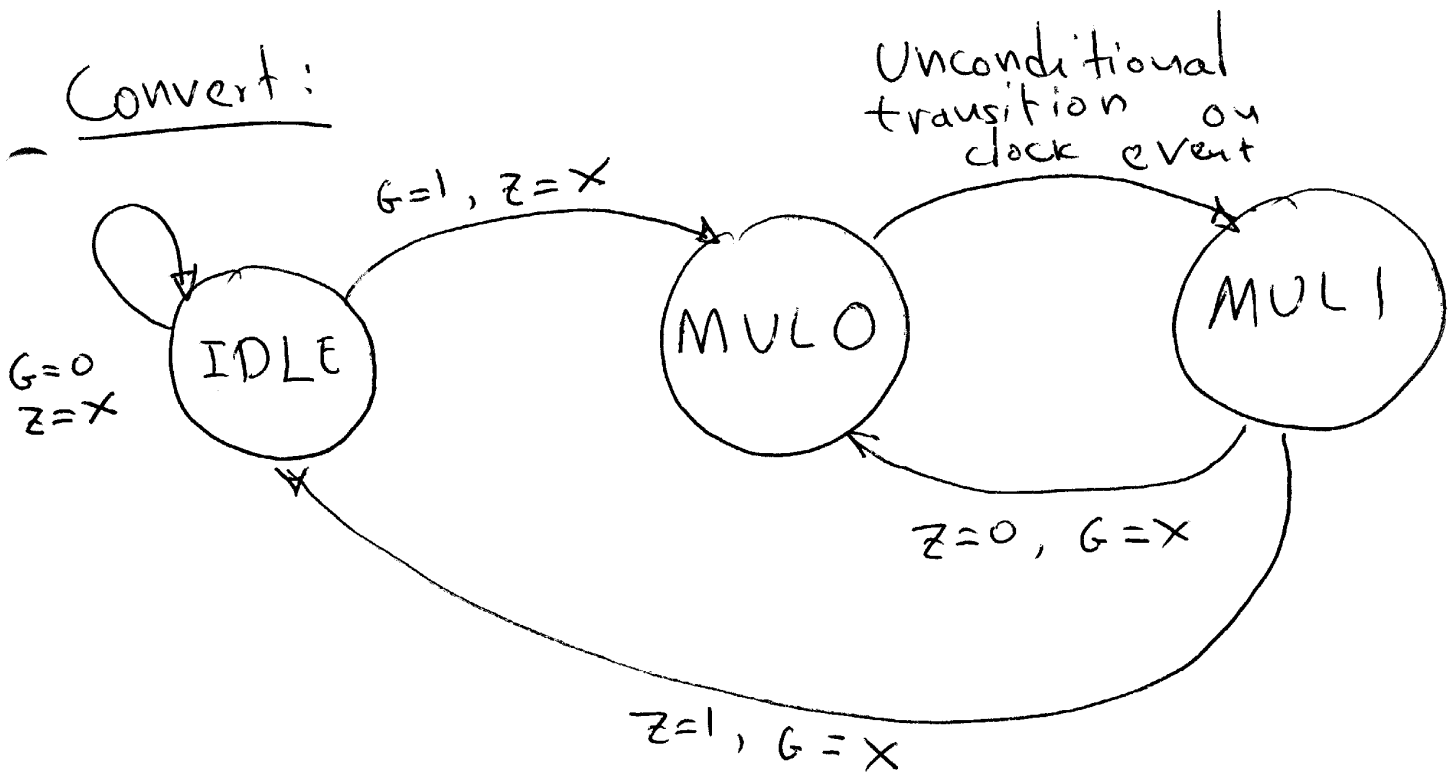
3 states: IDLE, MULO, MULI

2 inputs: G, Z

state assignment:

IDLE	=	00
MULO	=	01
MULI	=	10

Convert:



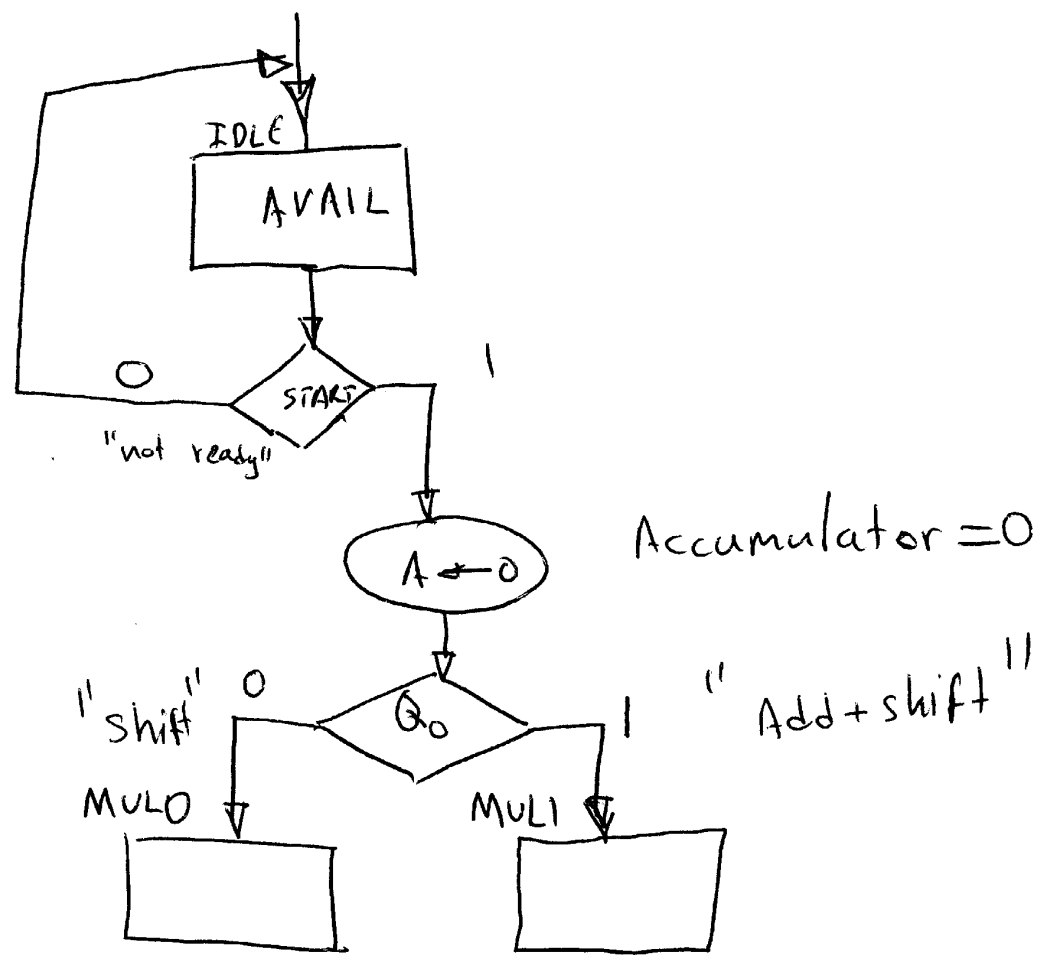
State Table:

Present State	Inputs		Next state
	G	Z	
IDLE: 00	0	X	IDLE: 00
	1	X	MULO: 01
MULO: 01	X	X	MUL1: 10
MUL1: 10	X	0	MULO: 01
	X	1	IDLE: 00
- 11	X	X	X

We can now implement the ASM's state transitions.

Timing.

consider:



States: 3: IDLE, MULO, MULI

1 register A

2 Inputs  $Q_0$ , START

1 output: AVAIL.

Problem:

start A to 1,

$Q_0 = 1$

state = IDLE,  $A = (034)_{16}$

⇒ Show A, state, output for 3 periods

