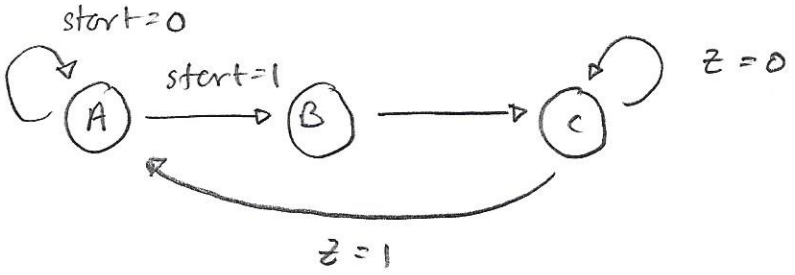


state A is the initial state

state B is the state where;

BR ← input A

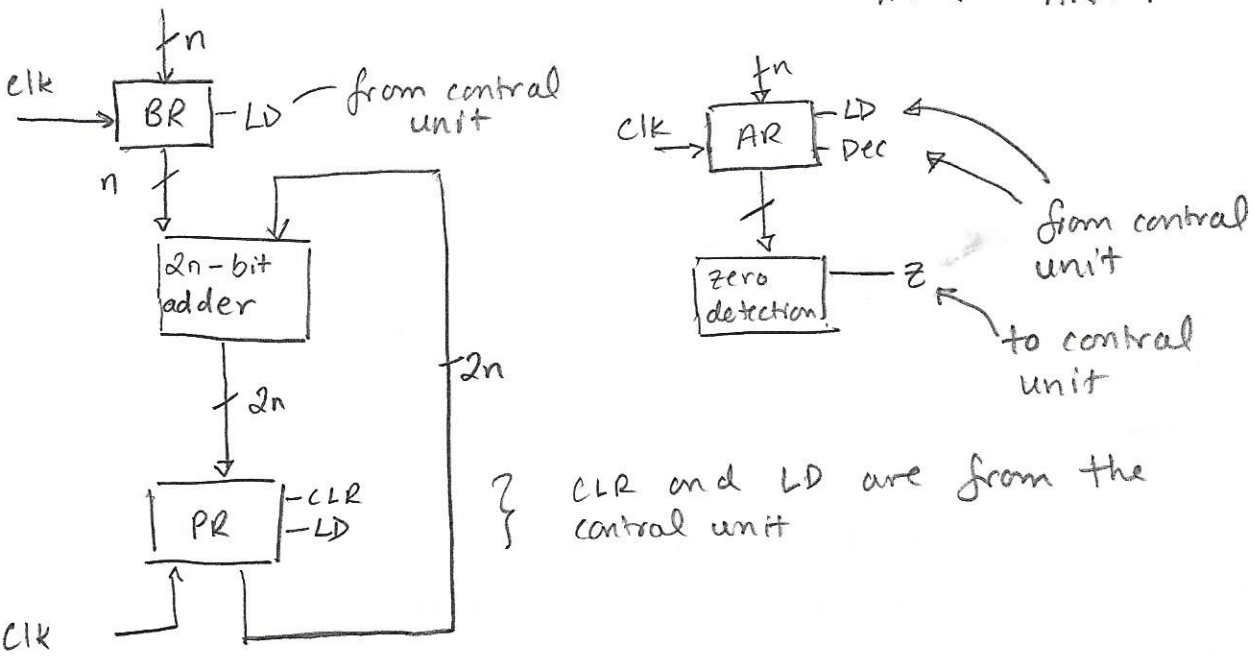
AR ← input B

PR ← 0

state C is the state where,

PR ← BR + PR

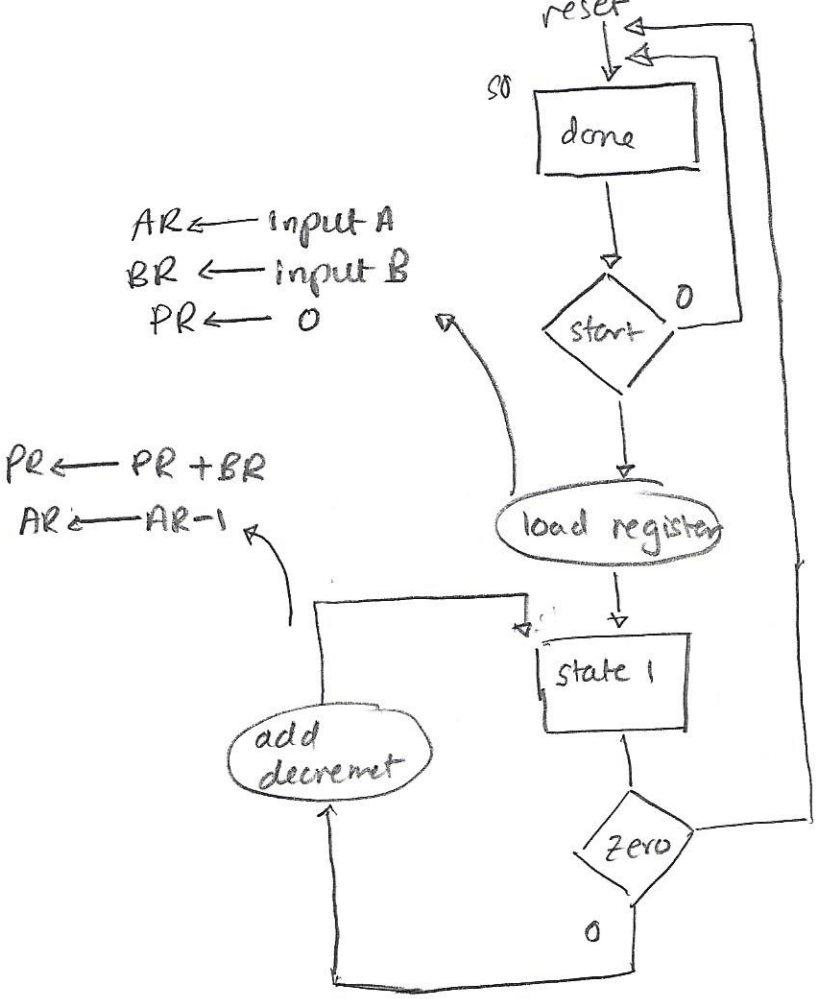
AR ← AR − 1



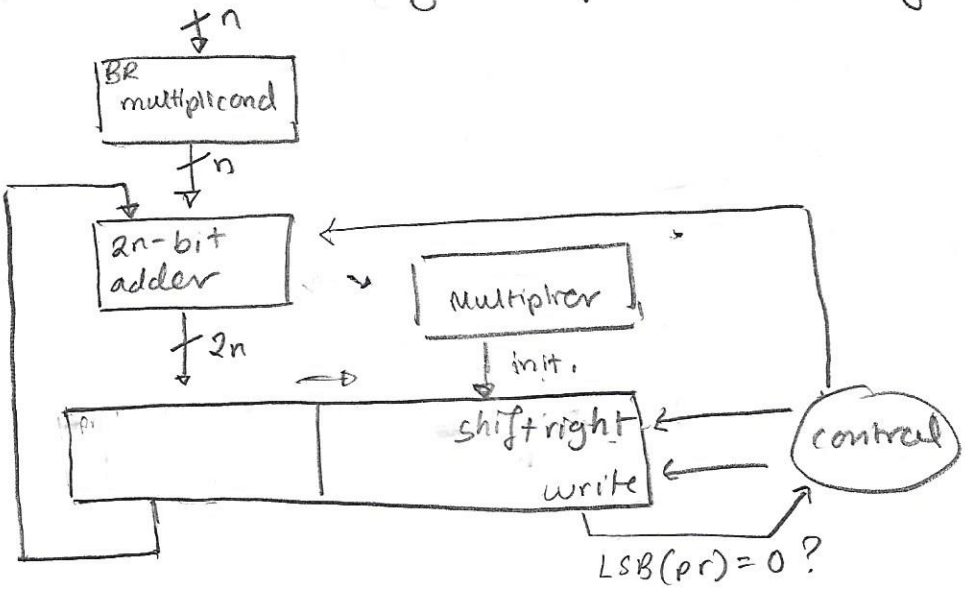
} CLR and LD are from the control unit

the internal architecture of the datapath consists of:

- a double width register to hold product → PR
- a register to hold the multiplicand → BR
- double-width parallel adder
- single-width parallel adder for decrementing the multiplier unit

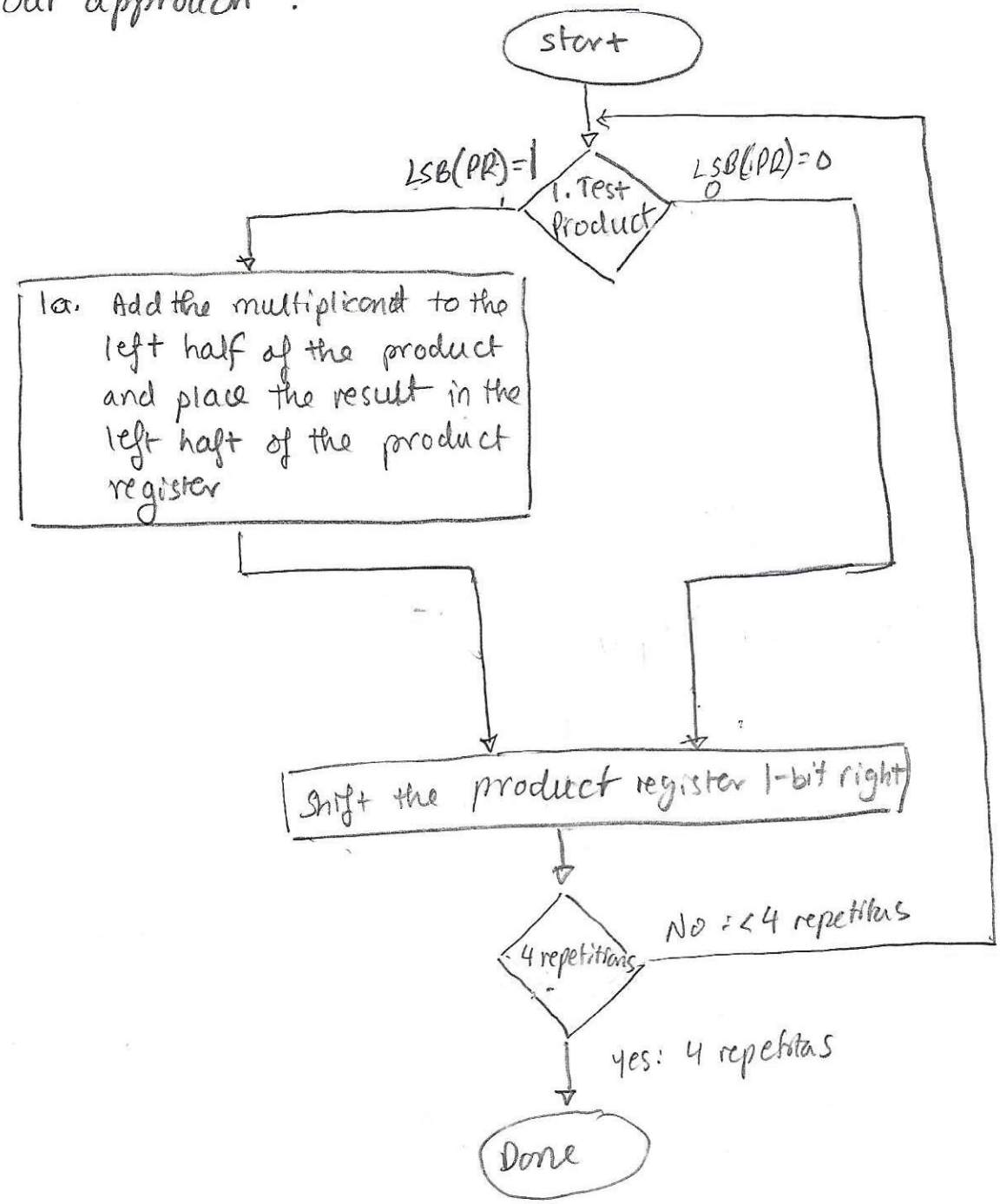


This is the expected ASM and state diagram for the multiplier. The components for our approach is as follows:



Multiplicand register, adder, and multiplier register are all 4 bits wide, with only the product register left a 8-bits. Multiplier is placed in the right half of the product register.

Our approach :



Iteration	Step	Multiplicand	Product
0	Initial values	0010	0000 0011
1	1a: 1 => product = product + multiplicand	0010	0010 0011
	2: Shift right product	0010	0001 0001
2	1a: 1 => product = product + multiplicand	0010	0001 1001
	2: Shift right product	0010	0000 1100
3	1: 0 => no operation	0010	0000 1100
	2: Shift right product	0010	0000 0110
4	1: 0 => no operation	0010	0000 1100
	2: shift right product	0010	0000 0110

check:

$$\begin{array}{r} 0011 \\ \times\ 0010 \\ \hline 0000\ 0110 \end{array}$$

$2^2 + 2^1 = 6$

$$\begin{array}{r} 3 \\ \times\ 2 \\ \hline 6 \end{array}$$

Iteration	Step	Multiplicand	Product
0	Initial values	0010	0000 0011
1	1a: 1 => product = product + multiplicand	0010	0010 0011
	2: Shift right product	0010	0001 0001
2	1a: 1 => product = product + multiplicand	0010	0011 0001
	2: Shift right product	0010	0001 1000
3	1: 0 => no operation	0010	0001 1000
	2: Shift right product	0010	0000 1100
4	1: 0 => no operation	0010	0000 1100
	2: shift right product	0010	0000 0110

check:

$$\begin{array}{r} 0011 \\ \times\ 0010 \\ \hline 0000\ 0110 \end{array}$$

$2^2 + 2^1 = 6$

$$\begin{array}{r} 3 \\ \times\ 2 \\ \hline 6 \end{array}$$