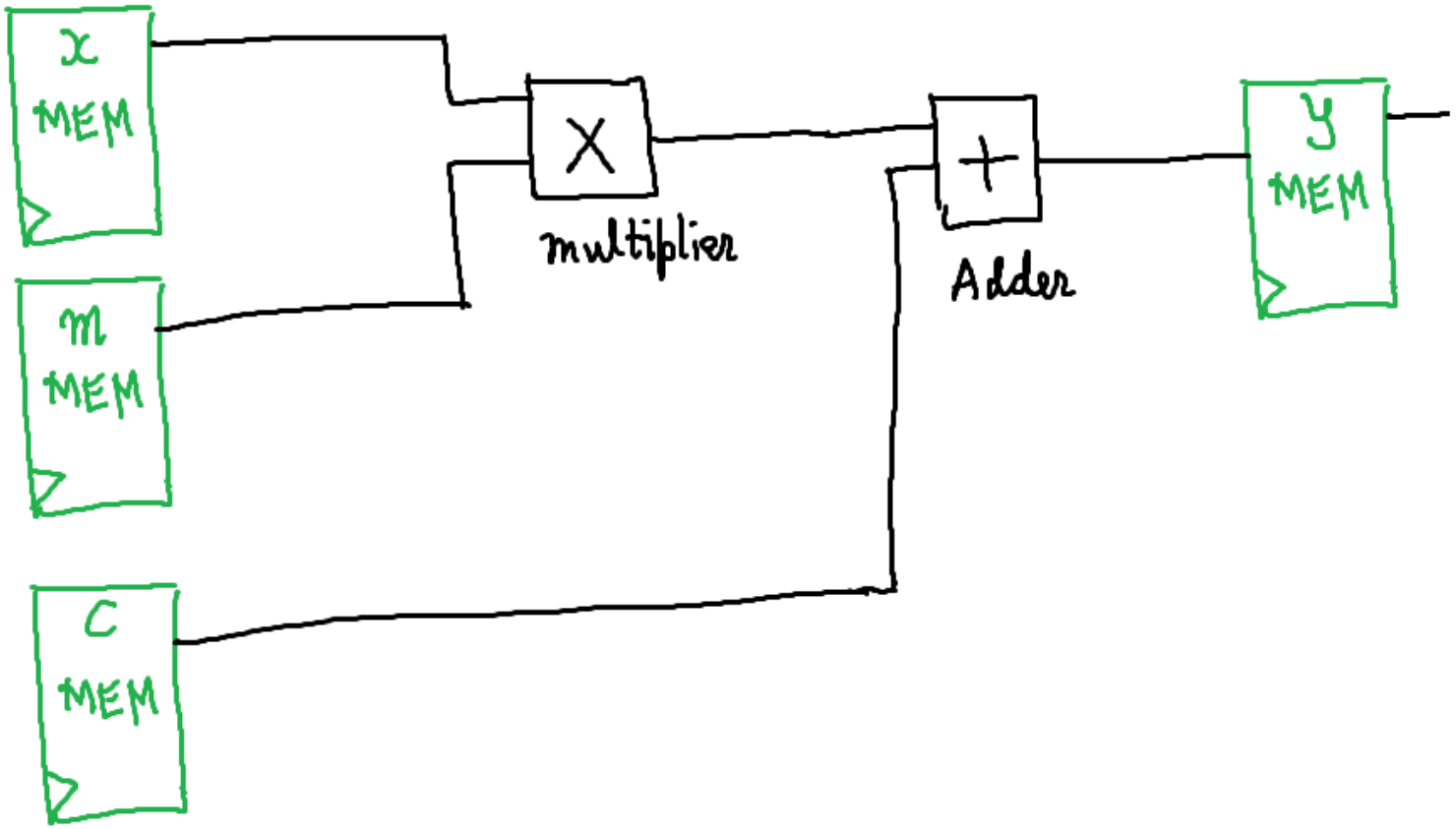
Pipelining and Parallel Processing

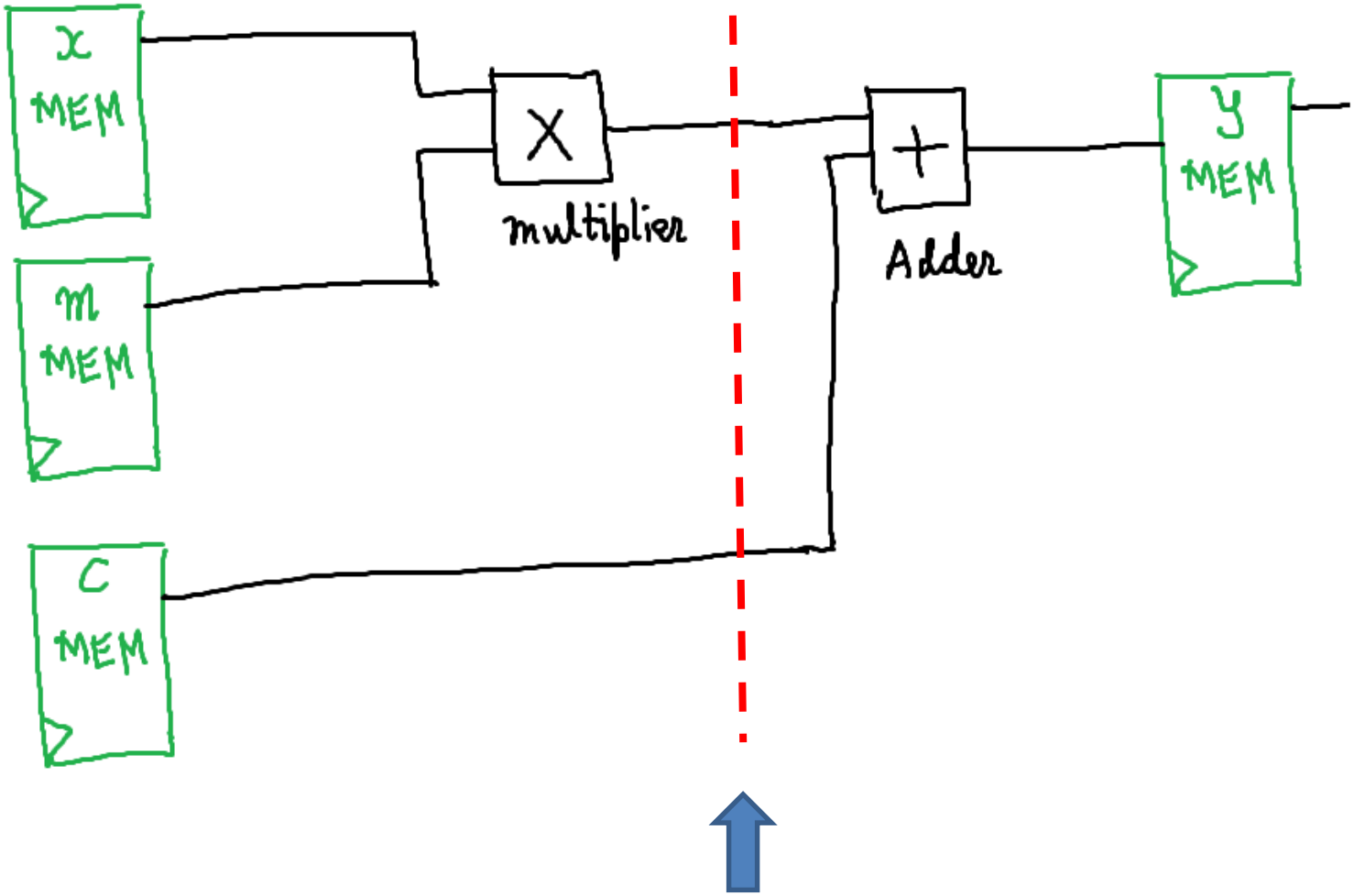
1-Stage (non-pipelined)

$$y_i = (m_i \times x_i) + c_i$$



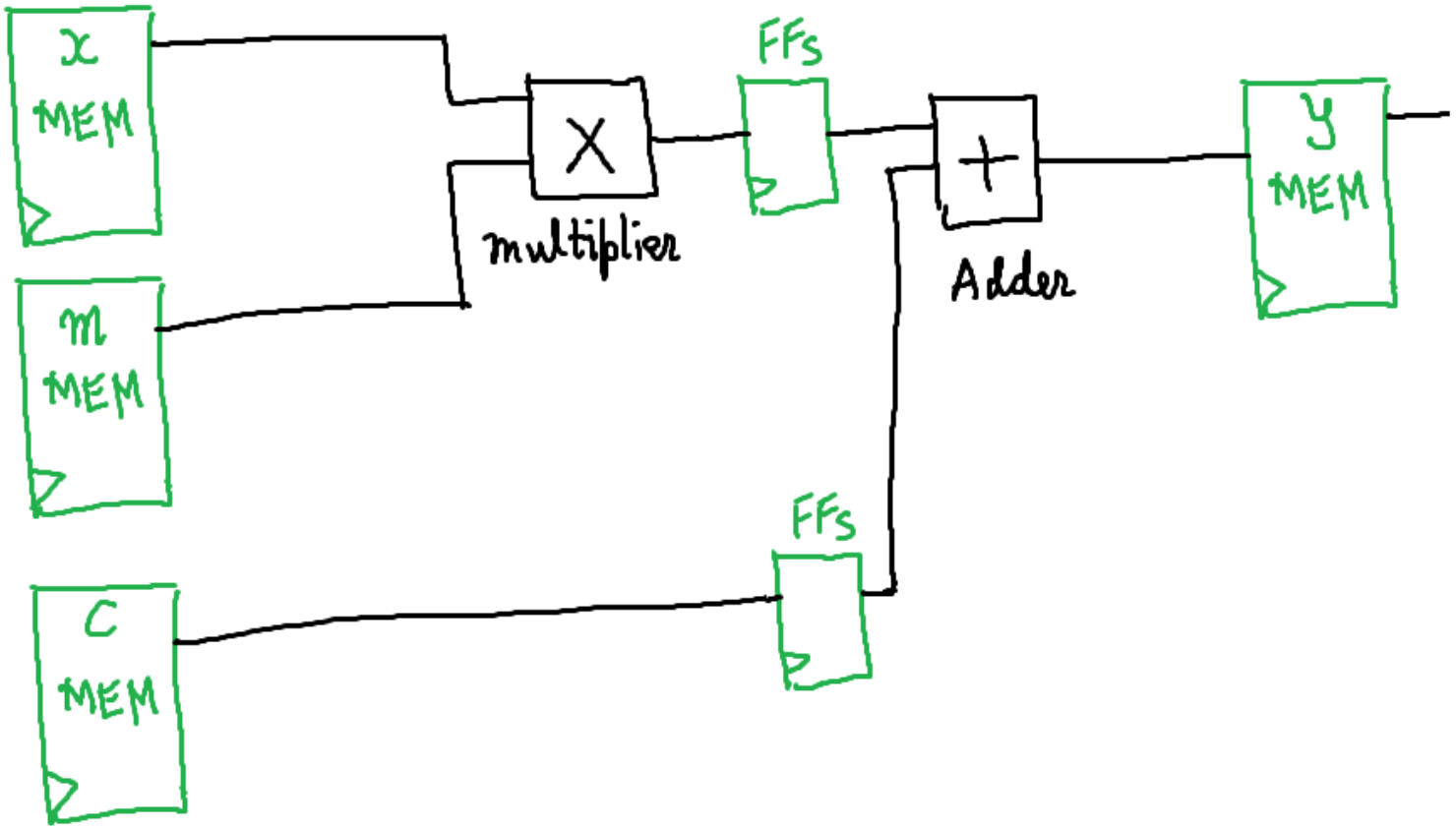
Decompose into more stages: How?

$$y_i = (m_i \times x_i) + c_i$$



2-Stage Pipelined

$$y_i = (m_i \times x_i) + c_i$$

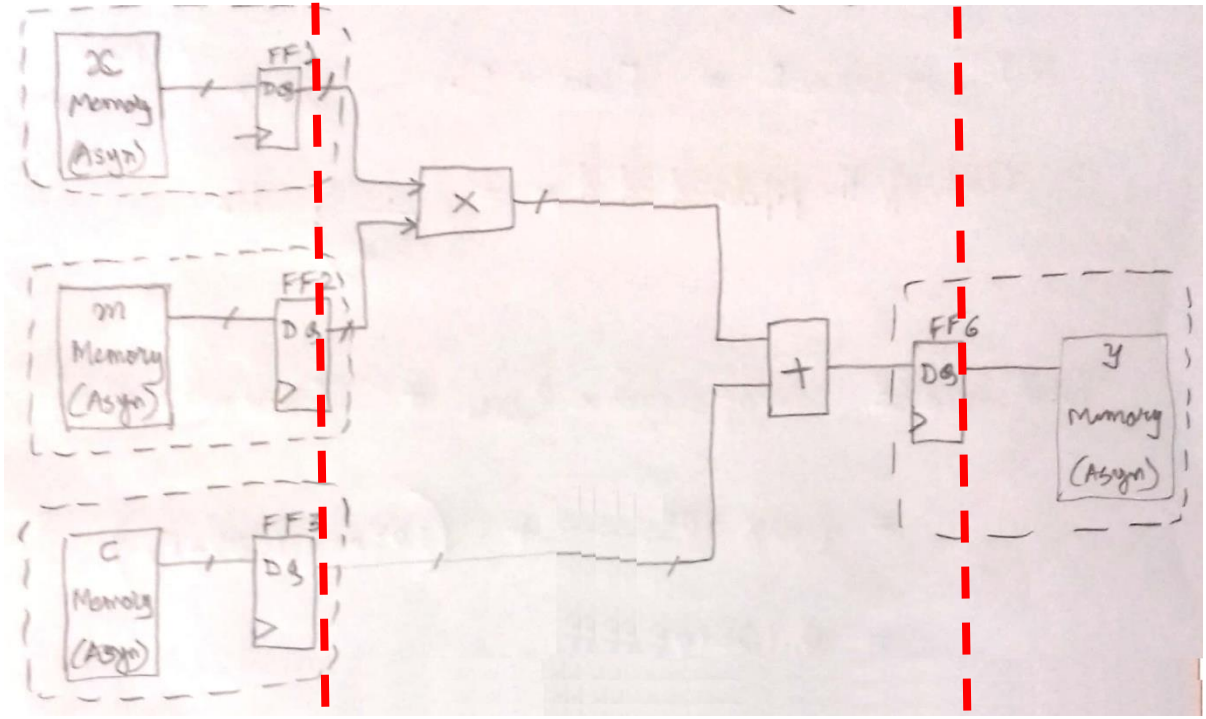


We want to implement $y_i = (m_i \times x_i) + c_i$

- **Un-pipelined design** (1 stage design)

Clock boundary

Data only moves from boundary to boundary when it gets clock edge



Clocked Element (Could be Clocked memory or FF or register

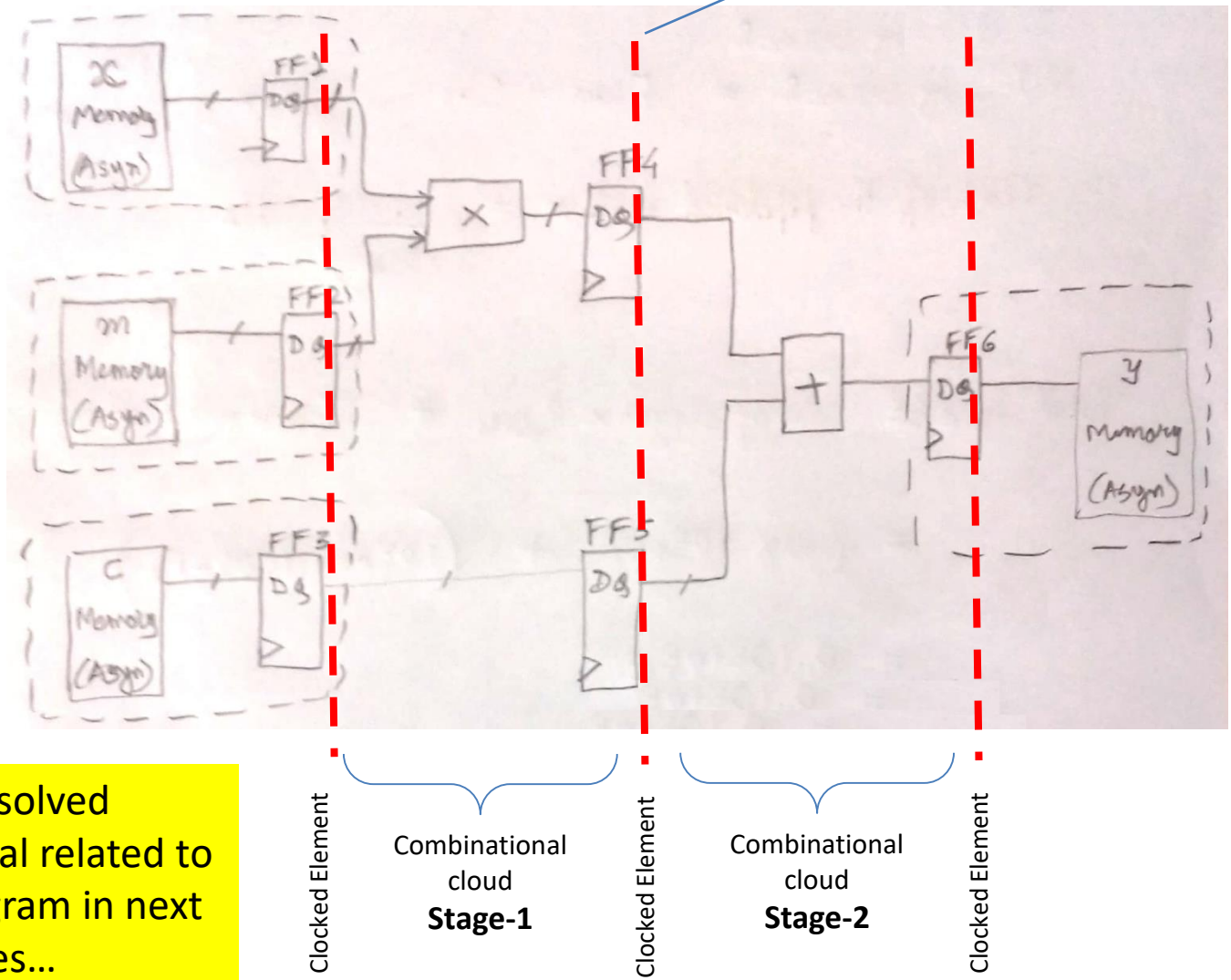
Combinational cloud

Clocked Element (Could be Clocked memory or FF or register

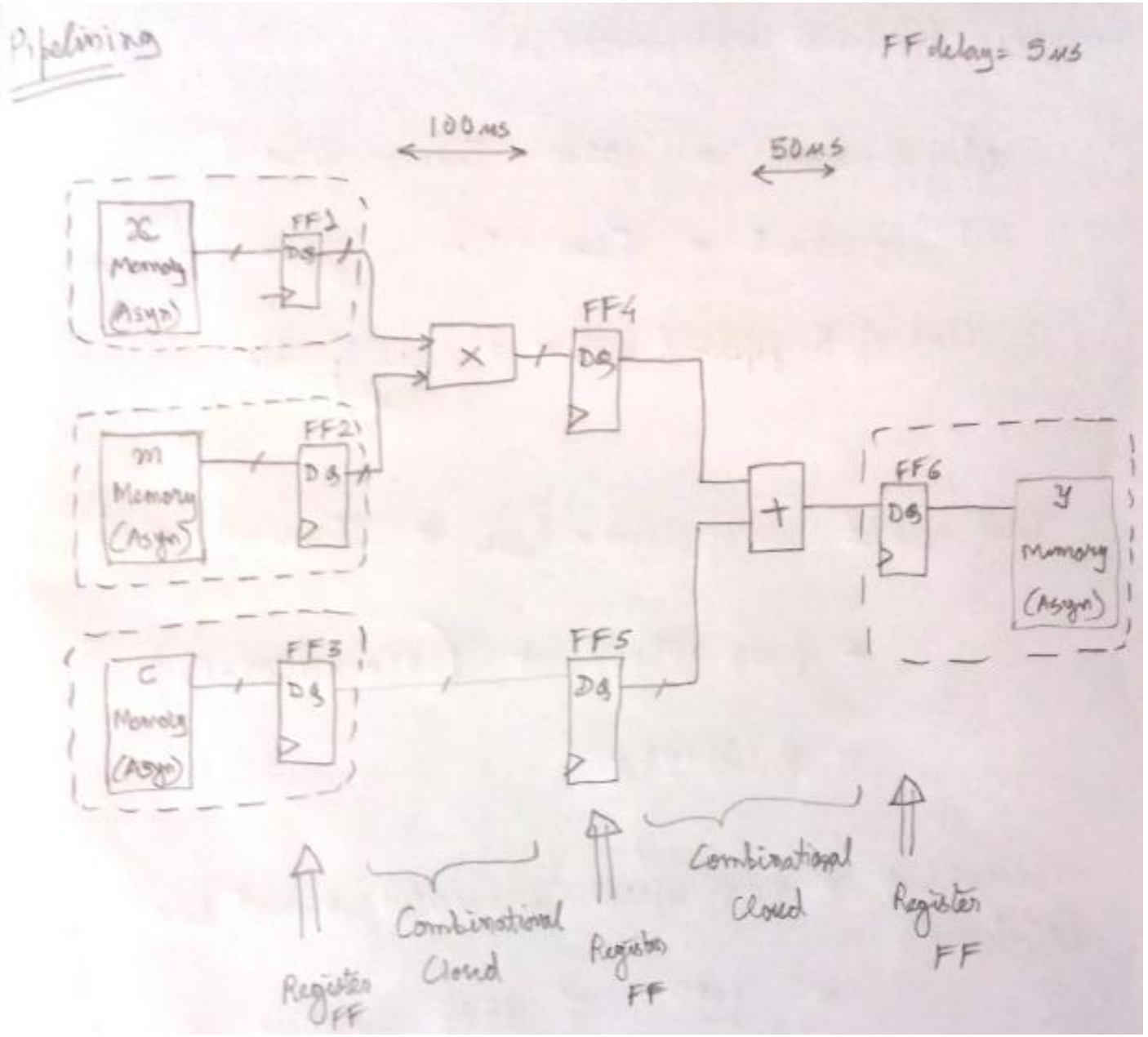
We want to implement $y_i = (m_i \times x_i) + c_i$

- **Pipelined design** (2 stage design)

We have deliberately cut the large combinational cloud into two by inserting this clocked elements / FFs



See the solved numerical related to this diagram in next few slides…


Pipelining
FF delay = 5 ns
100 ns
50 ns
x Memory (Asyn)
FF1
m Memory (Asyn)
FF2
C Memory (Asyn)
FF3
X
FF4
+
FF5
FF6
y Memory (Asyn)
DQ
Register FF
Combinational Cloud
Register FF
Combinational Cloud
Register FF

Let's say we have to do 1000 ($y = mx + c$) operations.

→ Unpipelined version: Delay = 100ns + 50ns = 150ns
for Combinational Cloud

↓ means

1 stage design delay = 150ns + 5ns = 155ns

Total delay = 155ns × 1000 = 0.155 s

2 stage pipelined implementation

delay of stage 1 = 100 ns + 5 ns = 105 ns

delay of stage 2 = 50 ns + 5 ns = 55 ns

} max = 105 ns = T_{clock}

∴ Clock of the pipelined system = $\left(\frac{1}{105n}\right)$ Hz

Total time ≅ no. of stages × T_{clock} + T_{clock} × (no of operations − 1)

= (2 × 105 ns) + (105 ns × (1000 − 1))

= 0.105105 s

Throughput (pipelined) = no. of operations executed per unit time

$$= \frac{1000}{0.105105} \cong 9514 \ \frac{\text{operations}}{\text{second}}$$

$$\% \text{ improvement in delay} = \left[\frac{(\text{total time}_{\text{unpipelined}} - \text{total time}_{\text{pipelined}})}{\text{total time}_{\text{unpipelined}}}\right] \times 100$$

$$\cong 32\ \%$$

How does data move in the pipe stages.

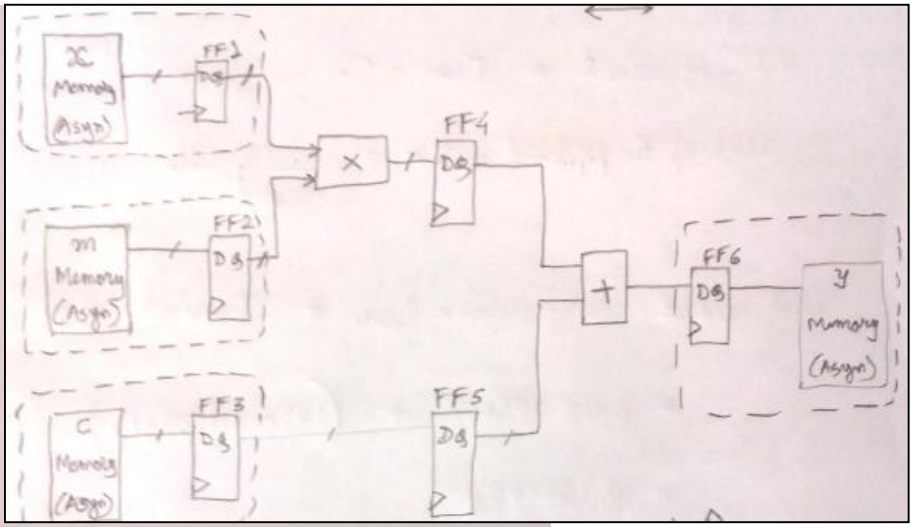
$x_1 = 1$ $c_1 = 2$ $m_1 = 0.5$

$x_2 = 3$ $c_2 = 1$ $m_2 = 0.9$

$x_3 = 5$ $c_3 = 10$ $m_3 = 2$

$x_4 = 7$ $c_4 = 13$ $m_4 = 1$

$x_5 = 9$ $c_5 = 5$ $m_5 = 0.8$



	Clock1	Clock2	Clock3	Clock4	Clock5	Clock6	Clock7	Clock8
FF_1 (x)	1	3	5	7	9	..	...	...
FF_2 (m)	0.5	0.9	2	1	0.8	...	...	..
FF_3 (c)	2	1	10	13	5	...	...	...
FF_4 (mx)	–	0.5	2.7	10	7	...	...	...
FF_5 (c)	–	2	1	10	13	...	...	...
FF_6 (y)	–	–	2.5	3.7	20	...	...	...