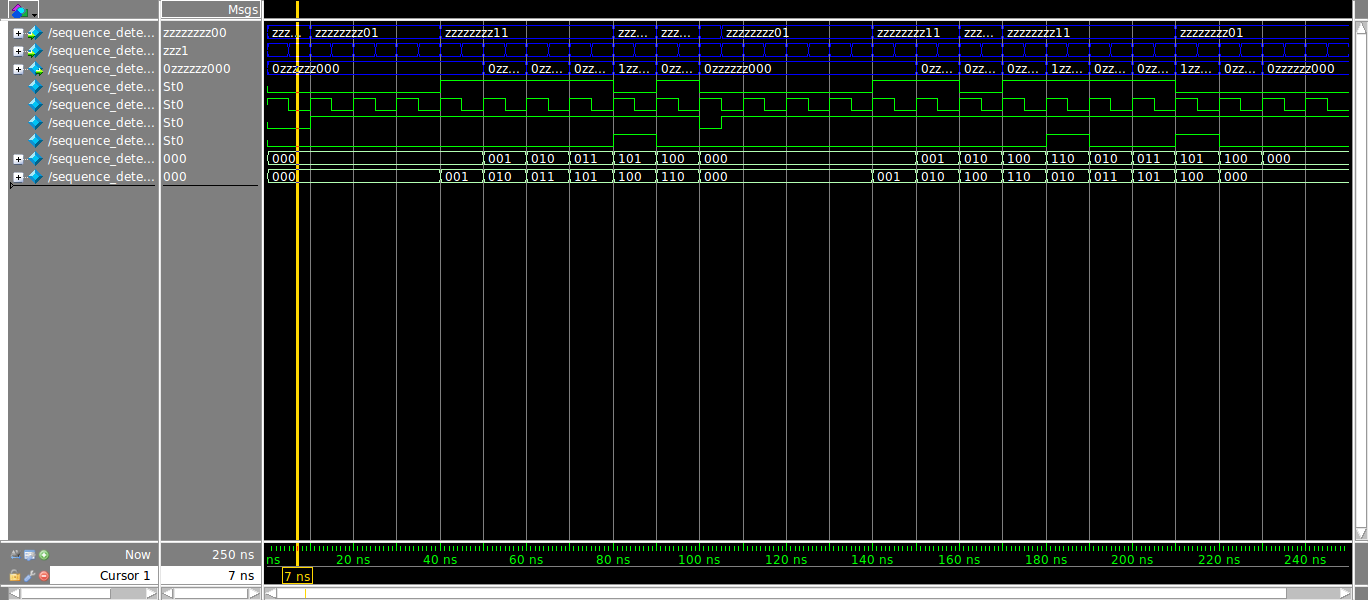
Part 1

2. resetn is asynchronous and active\_low. To reset we have to set resetn to 0 and clock from 0 to 1.

3.

|  |  |  |  |
| --- | --- | --- | --- |
| y\_Q | w | Y\_D | z |
| A | 0 | A | 0 |
| A | 1 | B | 0 |
| B | 0 | A | 0 |
| B | 1 | C | 0 |
| C | 0 | E | 0 |
| C | 1 | D | 0 |
| D | 0 | E | 0 |
| D | 1 | F | 0 |
| E | 0 | A | 0 |
| E | 1 | G | 0 |
| F | 0 | E | 1 |
| F | 1 | F | 1 |
| G | 0 | A | 1 |
| G | 1 | C | 1 |

4.

Part 2

2.

Load A to register A

data\_in = A, ld\_alu\_out = 0, ld\_a = 1, ld\_b = 0, ld\_c = 0, ld\_x = 0, ld\_r = 0

alu\_select\_a = 00, alu\_select\_b = 00, alu\_op = 0

registerA = A, registerB = 8’d0, registerC = 8’d0, registerX = 8’d0

Load B to register B

data\_in = B, ld\_alu\_out = 0, ld\_a = 0, ld\_b = 1, ld\_c = 0, ld\_x = 0, ld\_r = 0

alu\_select\_a = 00, alu\_select\_b = 00, alu\_op = 0

registerA = A, registerB = B, registerC = 8’d0, registerX = 8’d0

Load C to register C

data\_in = C, ld\_alu\_out = 0, ld\_a = 0, ld\_b = 0, ld\_c = 1, ld\_x = 0, ld\_r = 0

alu\_select\_a = 00, alu\_select\_b = 00, alu\_op = 0

registerA = A, registerB = B, registerC = C, registerX = 8’d0

Load X to register X

data\_in = X, ld\_alu\_out = 0, ld\_a = 0, ld\_b = 0, ld\_c = 0, ld\_x = 1, ld\_r = 0

alu\_select\_a = 00, alu\_select\_b = 00, alu\_op = 0

registerA = A, registerB = B, registerC = C, registerX = X

Multiply AX and store in registerA

ld\_alu\_out = 1, ld\_a = 1, ld\_b = 0, ld\_c = 0, ld\_x = 0, ld\_r = 0

alu\_select\_a = 00, alu\_select\_b = 11, alu\_op = 1

registerA = AX, registerB = B, registerC = C, registerX = X

Multiply AX² and store in registerA

ld\_alu\_out = 1, ld\_a = 1, ld\_b = 0, ld\_c = 0, ld\_x = 0, ld\_r = 0

alu\_select\_a = 00, alu\_select\_b = 11, alu\_op = 1

registerA = AX², registerB = B, registerC = C, registerX = X

Multiply BX and store in registerB

ld\_alu\_out = 1, ld\_a = 0, ld\_b = 1, ld\_c = 0, ld\_x = 0, ld\_r = 0

alu\_select\_a = 01, alu\_select\_b = 11, alu\_op = 1

registerA = AX², registerB = BX, registerC = C, registerX = X

Add AX²+BX and store in registerB

ld\_alu\_out = 1, ld\_a = 0, ld\_b = 1, ld\_c = 0, ld\_x = 0, ld\_r = 0

alu\_select\_a = 00, alu\_select\_b = 01, alu\_op = 0

registerA = AX², registerB = AX²+BX, registerC = C, registerX = X

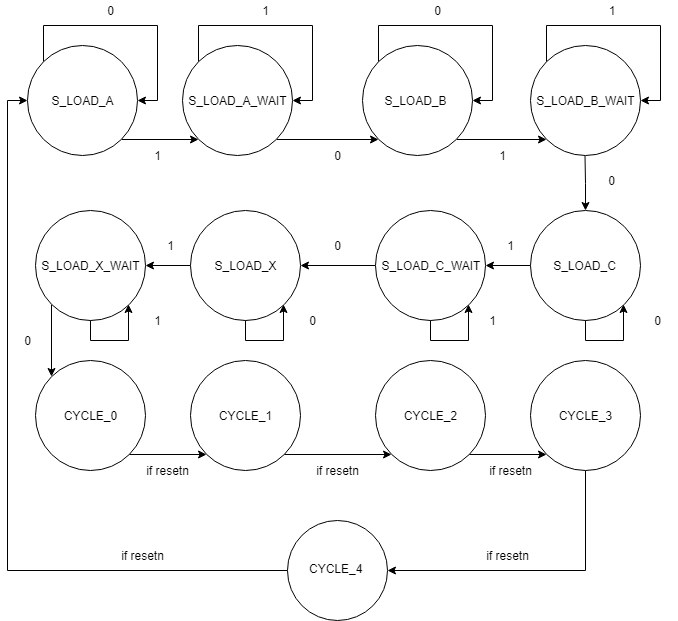
n

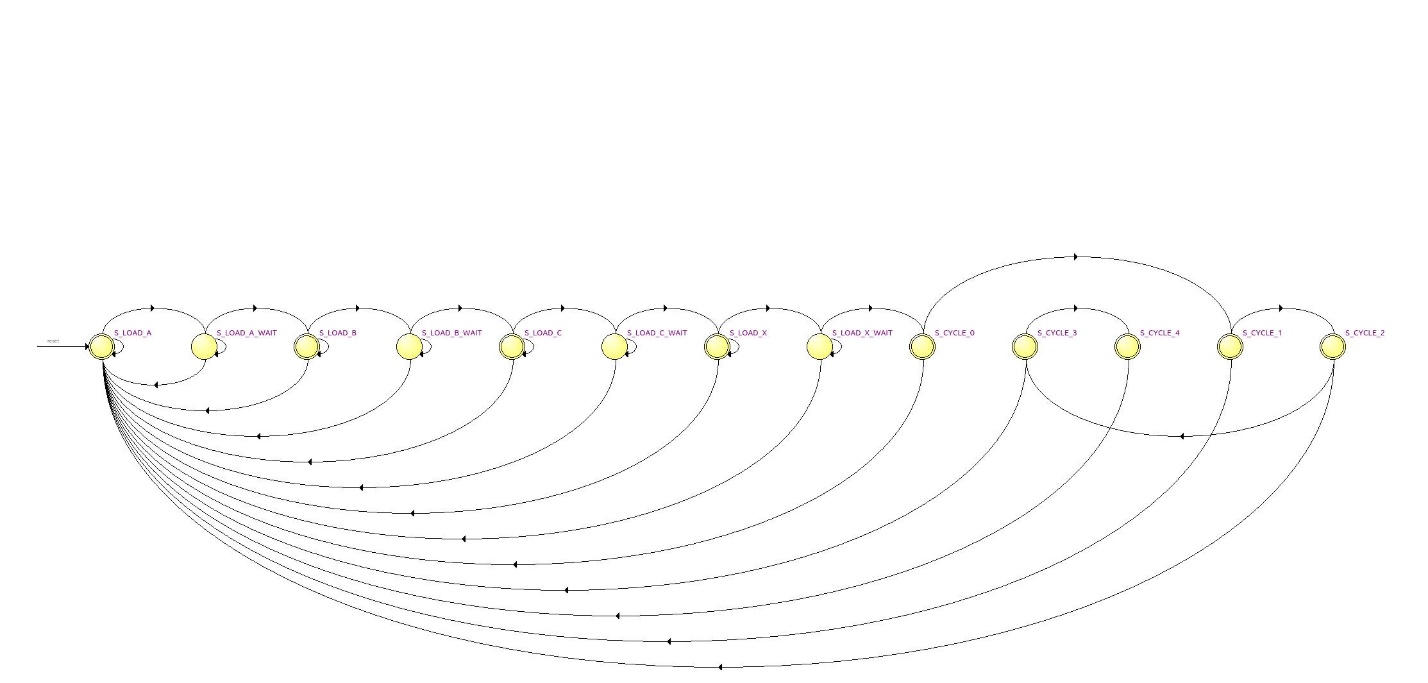
Add C to AX²+BX

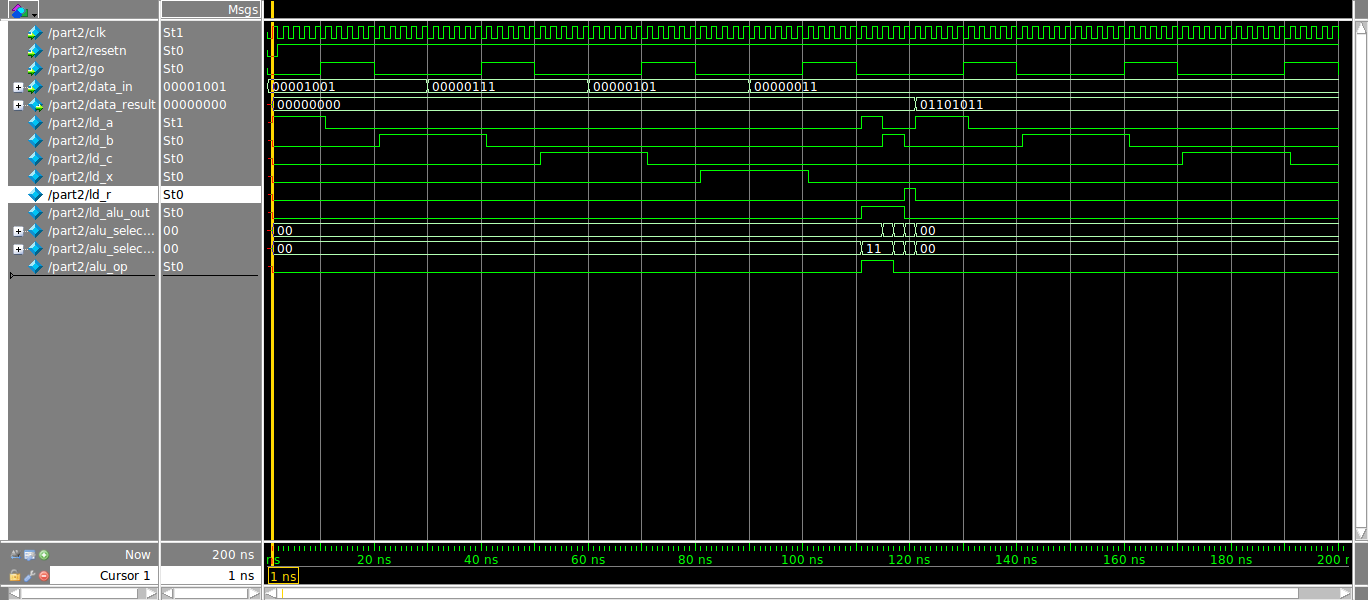
ld\_alu\_out = 0, ld\_a = 0, ld\_b = 0, ld\_c = 0, ld\_x = 0, ld\_r = 1

alu\_select\_a = 10, alu\_select\_b = 01, alu\_op = 0

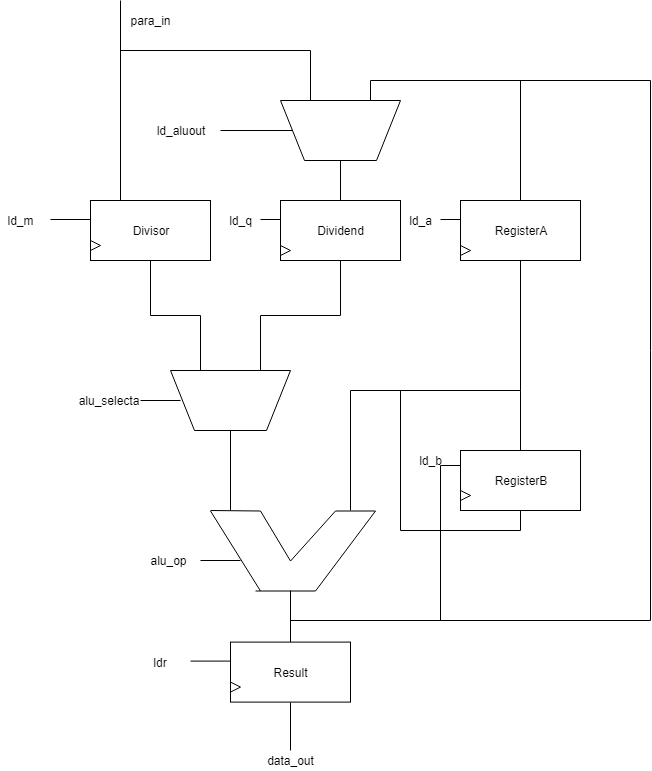
registerA = AX², registerB = AX²+BX, registerC = C, registerX = X

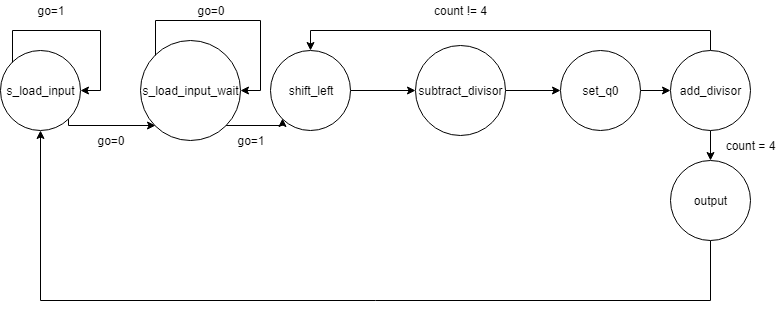
3.

4.

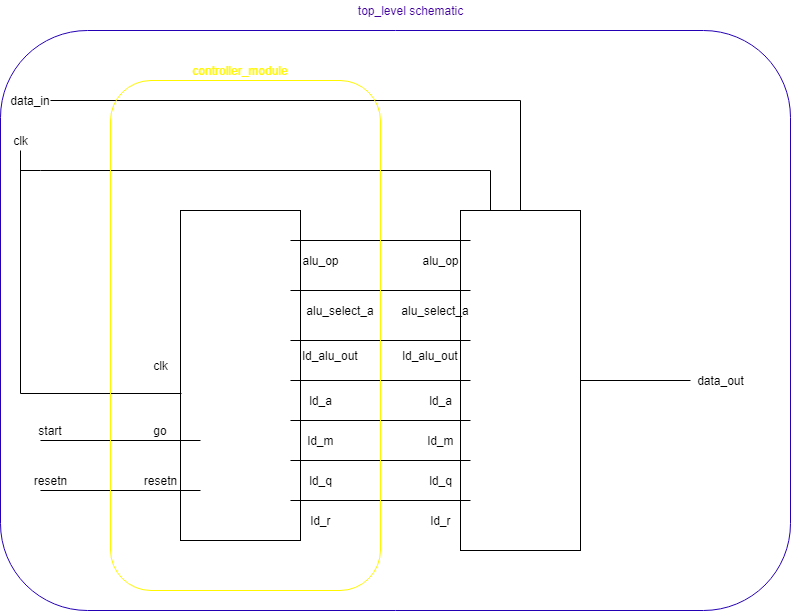
5.

Part3

1.

2.

3.



4.

