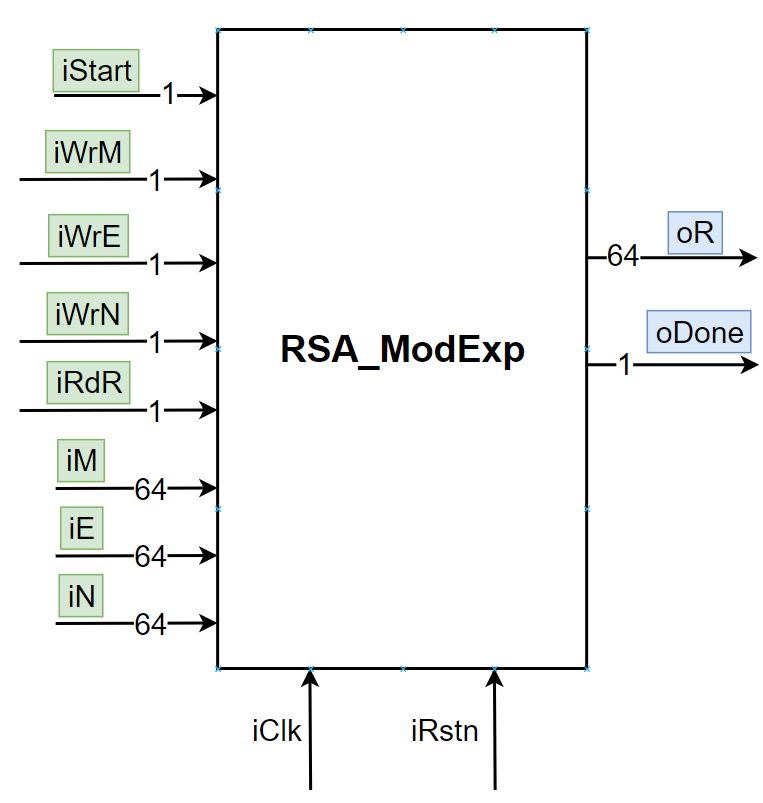
RSA

DIAGRAMS

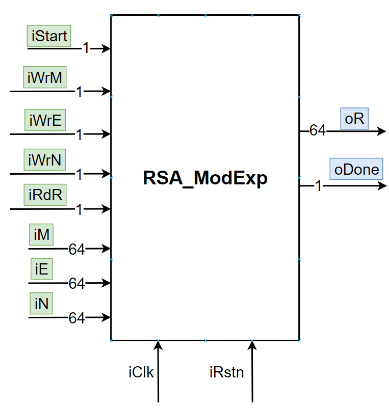
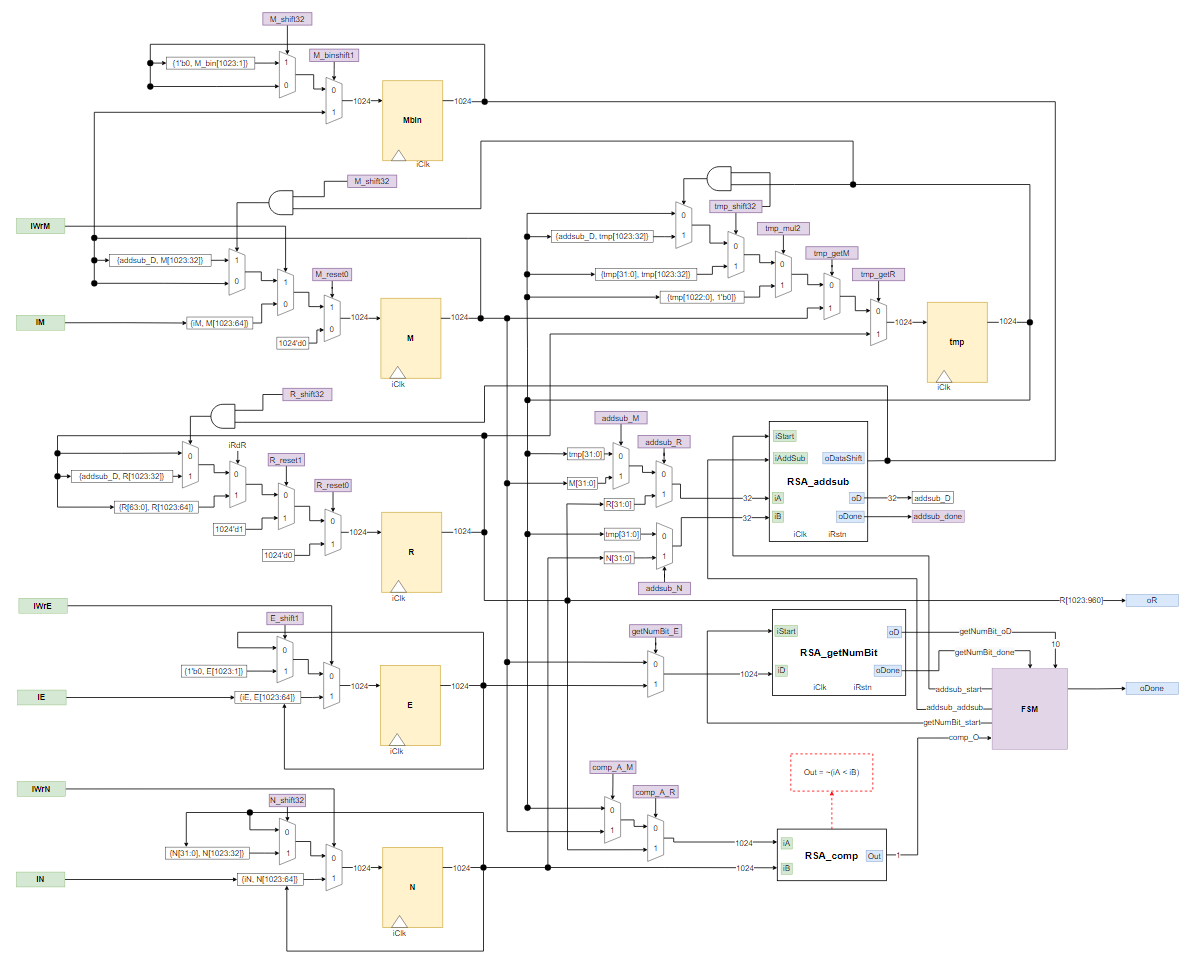
and STATE MACHINE

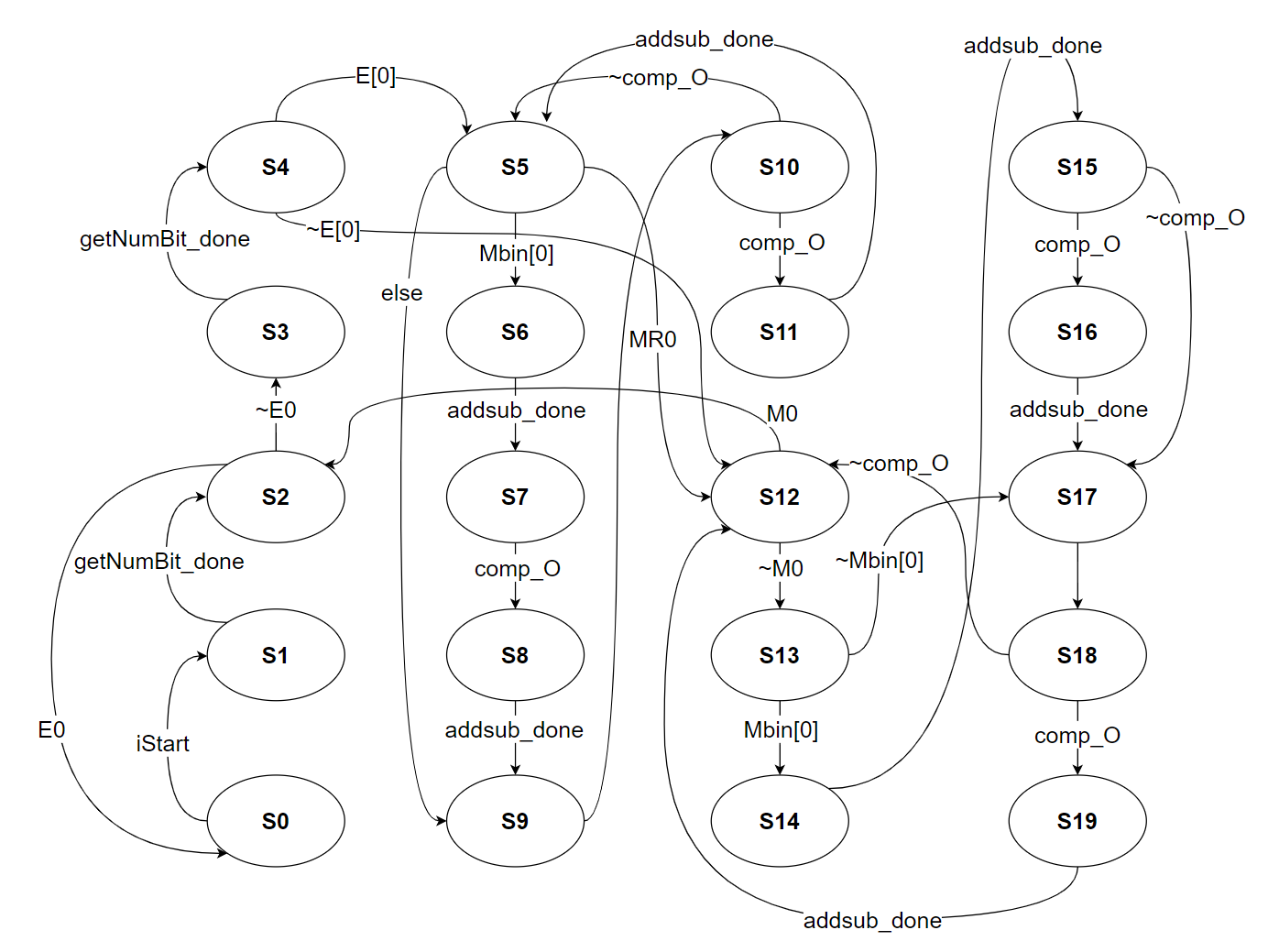


|  |  |  |  |
| --- | --- | --- | --- |
| PIN | DIR | WIDTH | Description |
| Control signals | | | |
| iClk | Input | 1 | Clock |
| iRstn | Input | 1 | Reset low |
| iStart | Input | 1 | Start computing |
| iWrM | Input | 1 | Write M |
| iWrE | Input | 1 | Write E |
| iWrN | Input | 1 | Write N |
| iRdR | Input | 1 | Read Result |
| Input Data | | | |
| iM | Input | 64 | M |
| iE | Input | 64 | E |
| iN | Input | 64 | N |
| Output Data | | | |
| oR | Output | 64 | Result |
| oDone | Output | 1 | Finish computing |

|  |  |
| --- | --- |
| Submodule | File Name |
| RSA\_addsub | RSA\_addsub.v |
| RSA\_getNumBit | RSA\_getNumBit.v |
| RSA\_comp | RSA\_comp.v |

|  |  |
| --- | --- |
| Name | File Name |
| RSA\_ModExp | RSA\_ModExp.v |





Finite State Machine

FSM Signals

**RESET**

oDone = 1’b0

State = 5’d0

M\_reset0 = 1’b0

M\_shift32 = 1’b0

Mbin\_update = 1’b0

Mbin\_shift1 = 1’b0

E\_shift1 = 1’b0

N\_shift32 = 1’b0

R\_reset0 = 1’b0

R\_reset1 = 1’b0

R\_shift32 = 1’b0

tmp\_getR = 1’b0

tmp\_getM = 1’b0

tmp\_mul2 = 1’b0

tmp\_shift32 = 1’b0

tmp\_shift32\_update = 1’b0

addsub\_start = 1’b0

addsub\_addsub = 1’b0

getNumBit\_start = 1’b0

comp\_A\_R = 1’b0

comp\_A\_M = 1’b0

getNumBit\_E = 1’b0

addsub\_R = 1’b0

addsub\_M = 1’b0

addsub\_N = 1’b0

lenE = 11’d0

lenM = 11’d0

**STATE0**

if (iStart) begin

State = 5’d1

getNumBit\_start = 1’b1

getNumBit\_E = 1’b1

else

State = State

**STATE1**

getNumBit\_start = 1’b0

if(getNumBit\_done)

lenE = getNumBit\_oD

R\_reset1 = 1

State = 5’d2

else

State = State

**STATE2**

R\_reset1 = 1’b0

if (lenE = 11’b0)

oDone = 1’b1

State = 5’d0

else

getNumBit\_start = 1’b1

getNumBit\_E = 1’b0

State = 5’d3

**STATE3**

getNumBit\_start = 1’b0

if (getNumBit\_done)

lenM = getNumBit\_oD

lenMR = getNumBit\_oD

Mbin\_update = 1’b1

State = 5’d4

else

State = State

**STATE4**

E\_shift1 = 1’b1

if (E[0])

Mbin\_update = 1’b0

tmp\_getR = 1’b1

R\_reset0 = 1’b1

State = 5’d5

else

Mbin\_update = 1’b1

tmp\_getR = 1’b1

M\_reset0 = 1’b1

State = 5’d12

**STATE5**

E\_shift1 = 1’b0

tmp\_getR = 1’b0

R\_reset0 = 1’b0

if (lenMR = 11’d0)

Mbin\_update = 1’b1

tmp\_getM = 1’b1

M\_reset0 = 1’b1

State = 5’d12

else

Mbin\_shift1 = 1’b1

if(Mbin[0])

addsub\_start = 1’b1

addsub\_addusb = 1’b1

addsub\_R = 1’b1

addsub\_N = 1’b0

R\_shift32 = 1’b1

tmp\_shift32 = 1’b1

State = 5’d6

else

tmp\_mul2 = 1’b1

State = 5’d9

**STATE6**

Mbin\_shift1 = 1’b0

addsub\_start = 1’b0

if (addsub\_done)

R\_shift32 = 1’b0

tmp\_shift32 = 1’b0

comp\_A\_R = 1’b1

State = 5’d7

else

State = State

**STATE7**

if (comp\_O)

addsub\_start = 1’b1

addsub\_addsub = 1’b1

addsub\_R = 1’b1

addsub\_N = 1’b1

R\_shift32 = 1’b1

N\_shift32 = 1’b1

State = 5’d8

else

tmp\_mul2 = 1’b1

State = 5’d9

**STATE8**

addsub\_start = 1’b0

if (addsub\_done)

R\_shift32 = 1’b0

N\_shift32 = 1’b0

tmp\_mul2 = 1’b1

State = 5’d9

else

State = State

**STATE9**

Mbin\_shift1 = 1’b0

tmp\_mul2 = 1’b0

comp\_A\_R = 1’b0

comp\_A\_M = 1’b0

State = 5’d10

**STATE10**

if (comp\_O)

addsub\_start = 1’b1

addsub\_addsub = 1’b1

addsub\_R = 1’b0

addsub\_M = 1’b0

addsub\_N = 1’B1

tmp\_shift32\_update = 1’b1

N\_shift32 = 1’b1

State = 5’d11

else

lenMR = lenMR – 1’b1

State = 5’d5

**STATE11**

addsub\_start = 1’b0

if (addsub\_done)

tmp\_shift32\_update = 1’b0

N\_shift32 = 1’b0

lenMR = len MR – 1’b1

State = 5’d5

else

State = State

**STATE12**

E\_shift1 = 1’b0

Mbin\_update = 1’b0

tmp\_getM = 1’b0

M\_reset0 = 1’b0

if (lenM = 11’d0)

lenE = lenE – 1’b1

State = 5’d2

else

State = 5’d13

**STATE13**

Mbin\_shift1 = 1’b1

if (Mbin[0])

addsub\_start = 1’b1

addsub\_addsub = 1’b0

addsub\_R = 1’b0

addsub\_M = 1’b1

addsub\_N = 1’b0

M\_shift32 = 1’b1

tmp\_shift32 = 1’b1

State = 5’d14

else

tmp\_mul2 = 1’b1

State = 5’d17

**STATE14**

Mbin\_shift1 = 1’b0

addsub\_start = 1’b0

if (addsub\_done)

M\_shift32 = 1’b0

tmp\_shift32 = 1’b0

comp\_A\_R = 1’b0

comp\_A\_M = 1’b1

State = 5’d15

else

State = State

**STATE15**

if(comp\_O)

addsub\_start = 1’b1

addsub\_addsub = 1’b1

addsub\_R = 1’b0

addsub\_M = 1’b1

addsub\_N = 1’b1

M\_shift32 = 1’b1

N\_shift32 = 1’b1

State = 5’d16

else

tmp\_mul2 = 1’b1

State = 5’d17

**STATE16**

addsub\_start = 1’b0

if (addsub\_done)

M\_shift32 = 1’b0

N\_shift32 = 1’b0

tmp\_mul2 = 1’b1

State = 5’d17

else

State = State

**STATE17**

Mbin\_shift1 = 1’b0

tmp\_mul2 = 1’b0

comp\_A\_R = 1’b0

comp\_A\_M = 1’b0

State = 5’d18

**STATE18**

if (comp\_O)

addsub\_start = 1’b1

addsub\_addsub = 1’b1

addsub\_R = 1’b1

addsub\_M = 1’b0

addsub\_N = 1’b1

tmp\_shift32\_update = 1’b1

N\_shift32 = 1’b1

State = 5’d19

else

lenM = lenM – 1’b1

State = 5’d12

**STATE19**

addsub\_start = 1’b0

if (addsub\_done)

tmp\_shift32\_update = 1’b0

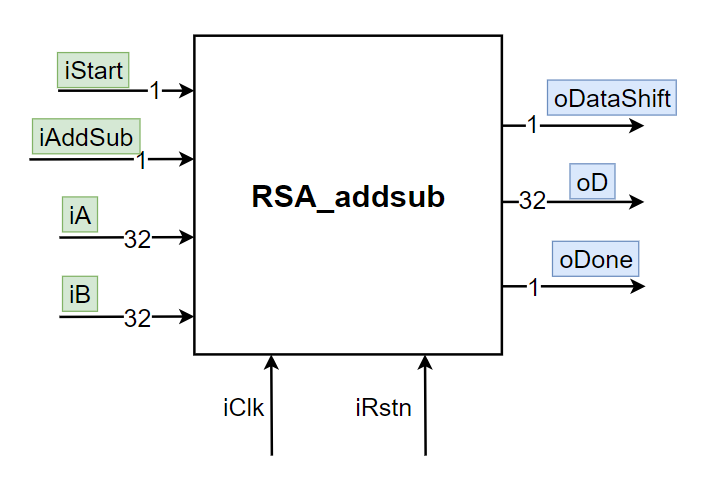
N\_shift32 = 1’b0

lenM = lenM – 1’b1

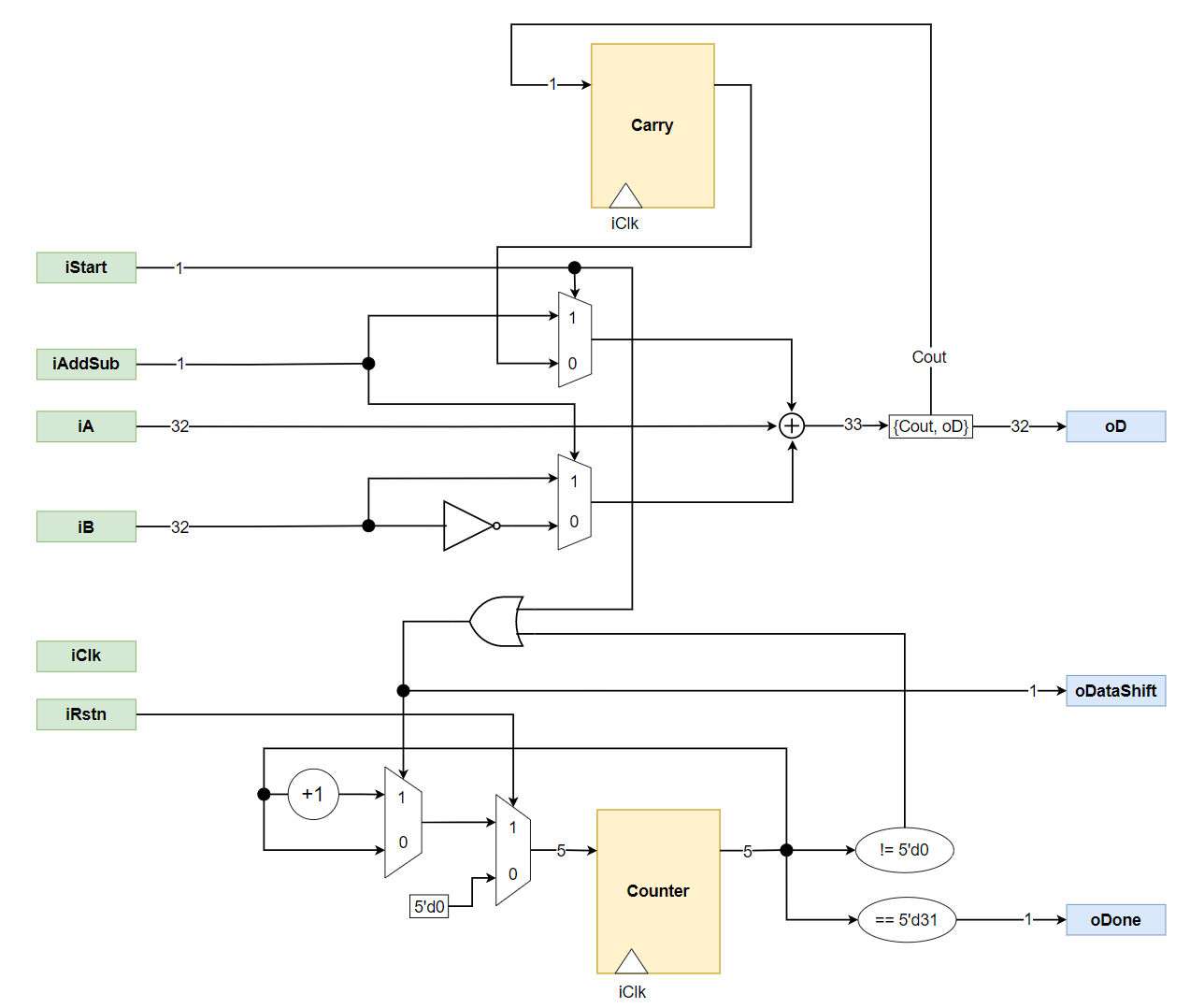
State = 5’d12

else

State = State



|  |  |
| --- | --- |
| Name | File Name |
| RSA\_addsub | RSA\_addsub.v |



|  |  |
| --- | --- |
| Name | File Name |
| RSA\_getNumBit | RSA\_getNumBit.v |

