

Figure: All basic gates

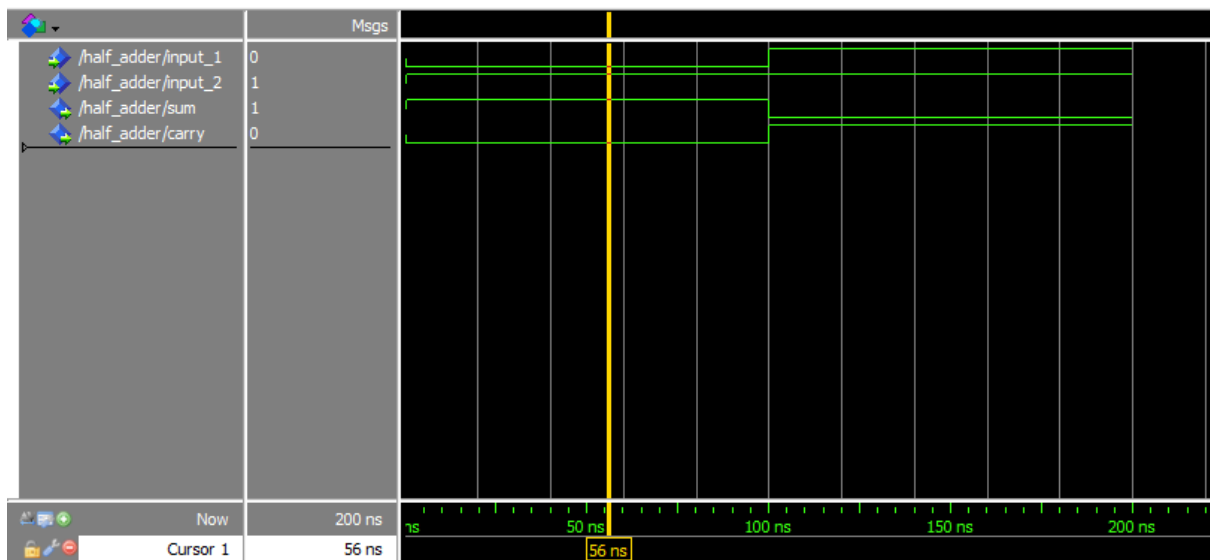


Figure: Half Adder

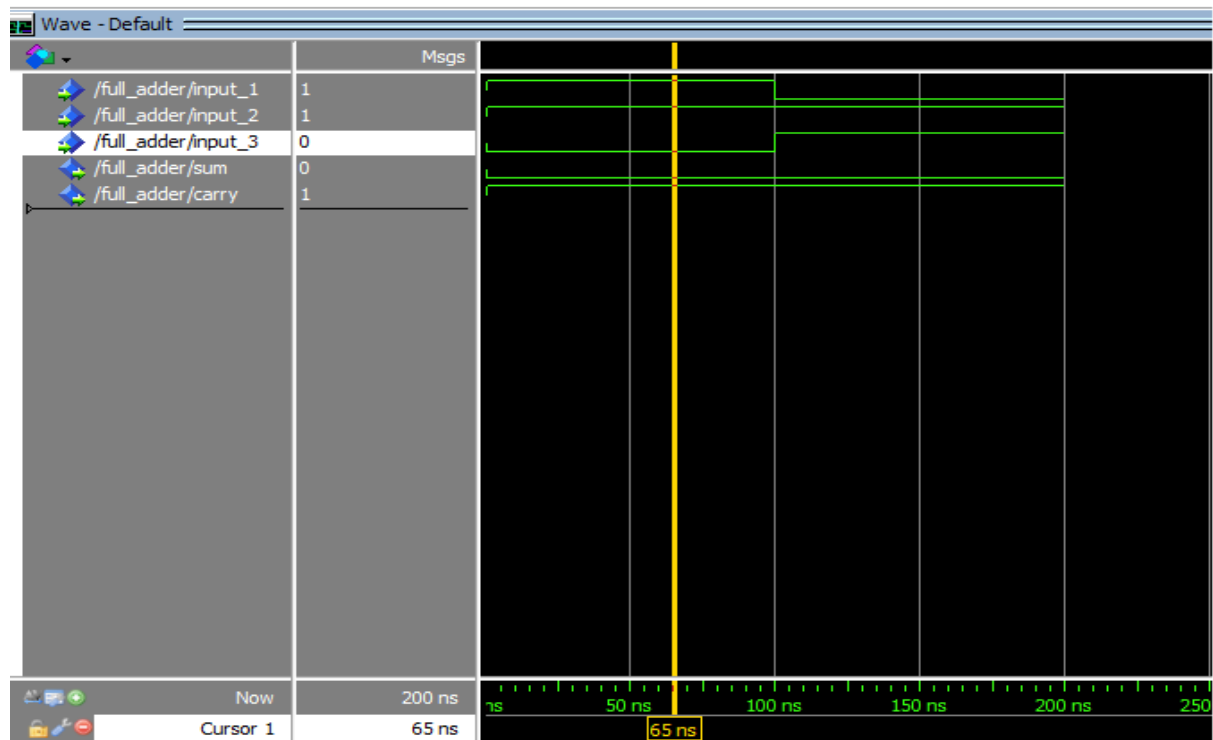


Figure: Full Adder

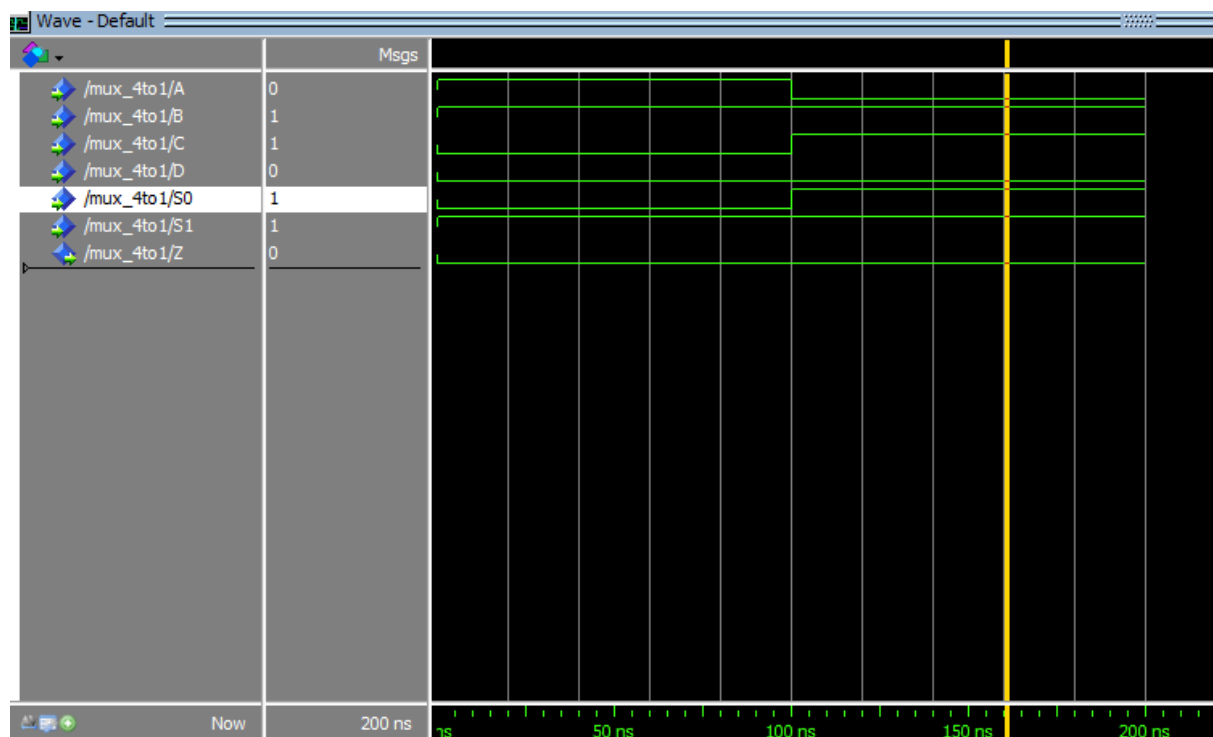


Figure: 4*1 Multiplexer

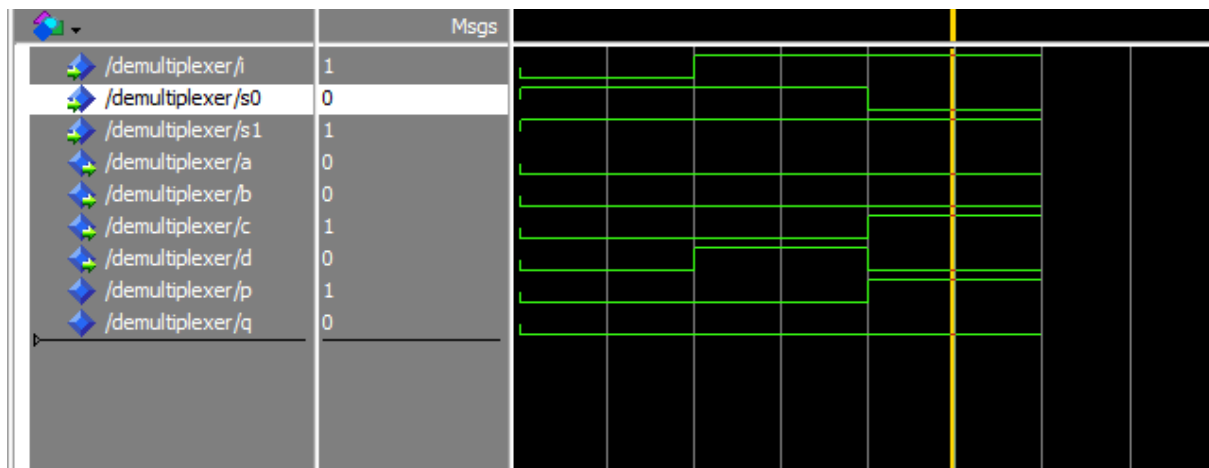


Figure: 1*4 Demultiplexer

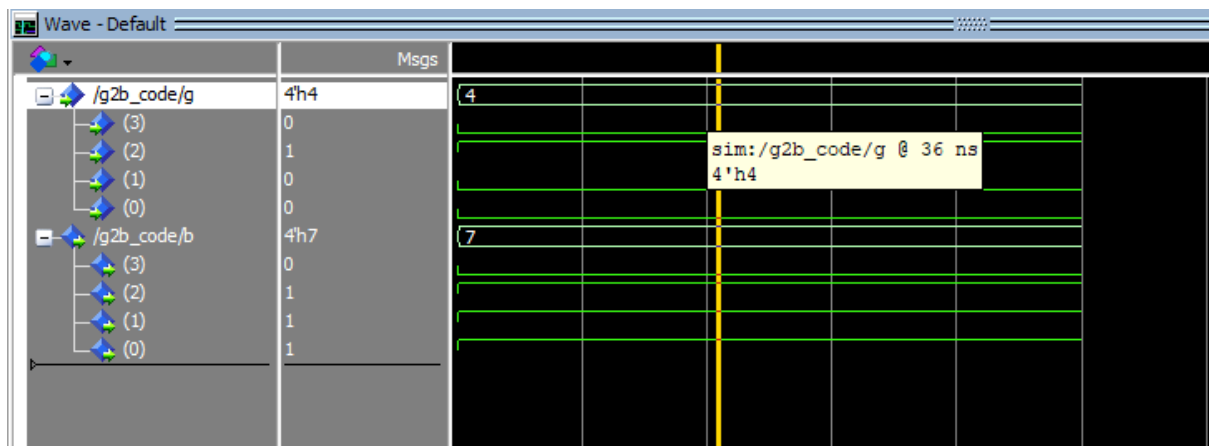


Figure: Gray to Binary code converter

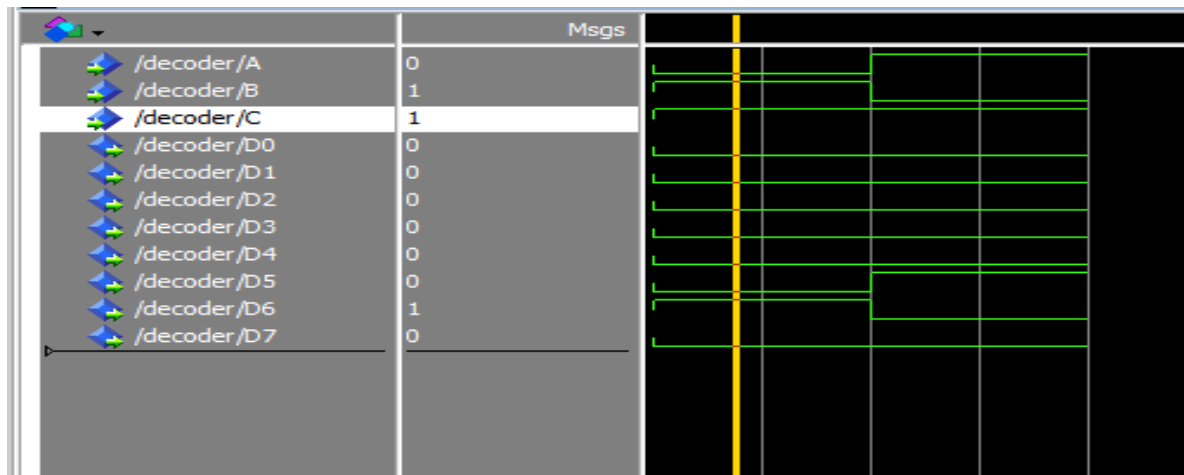


Figure: 3*8 Decoder



Figure: 8*3 Encoder

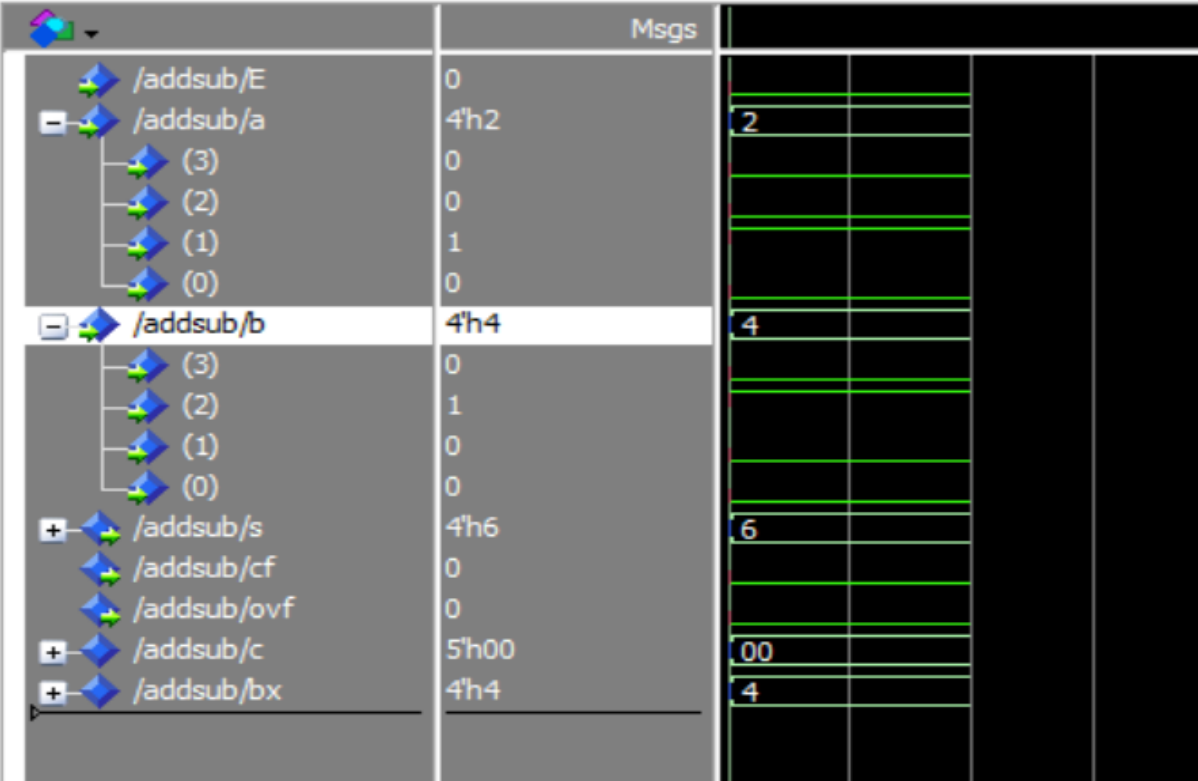


Figure: 2's Complement Adder-Subtractor

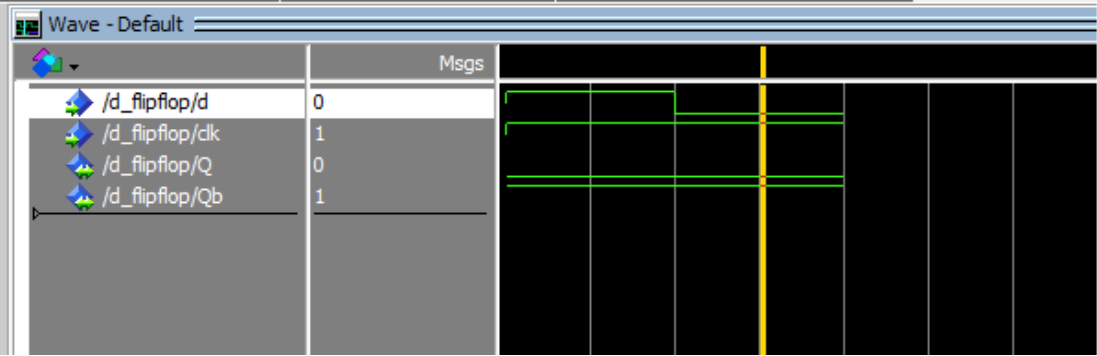


Figure: D flip-flop

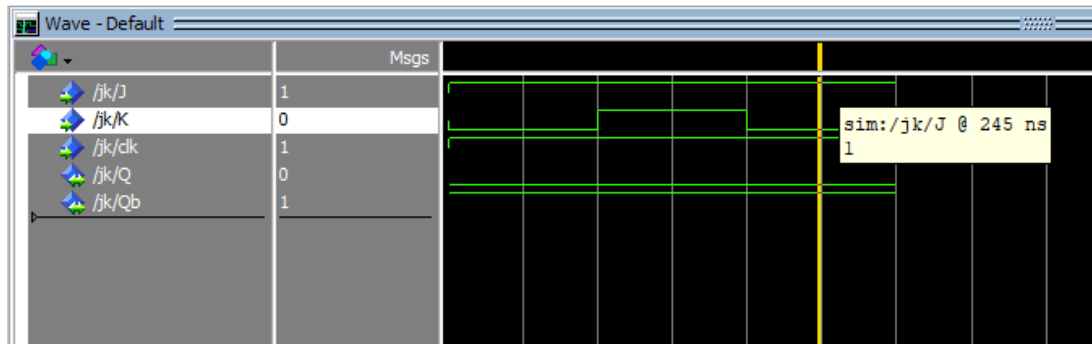


Figure: JK flip-flop

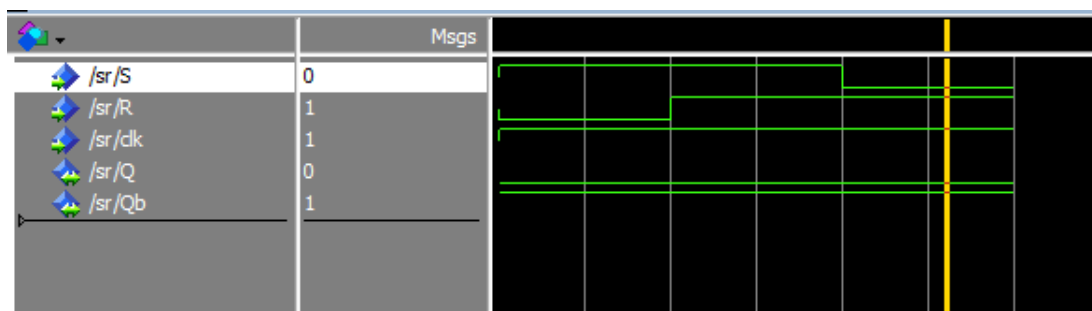


Figure: SR flip-flop

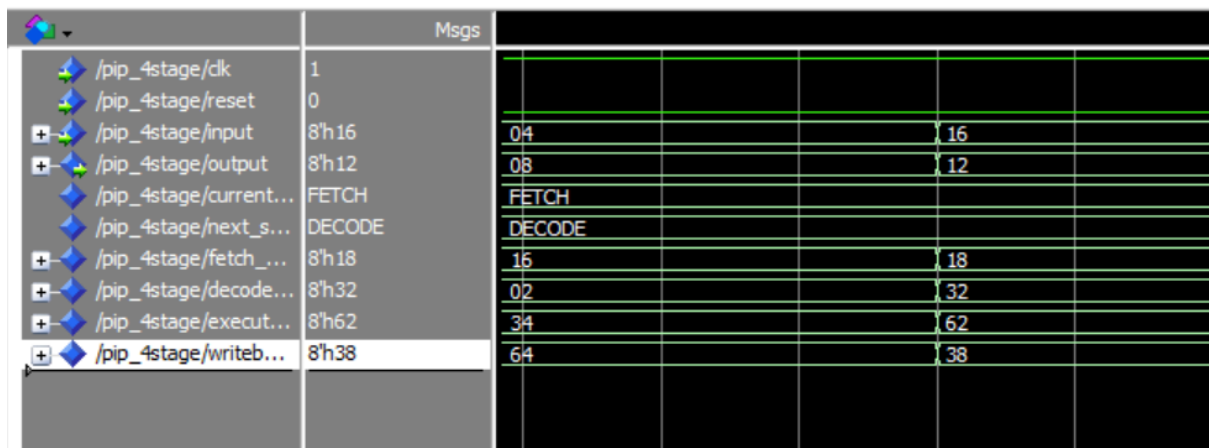


Figure: 4-stage pipelining

```

Kiran Joshi Sukubhattu
          SIGNED MAGNITUDE MULTIPLICATION ALGORITHM
Enter two numbers to multiply:
Both must be less than 16
Enter b: 7
Enter Q: 2

Expected product = 14
S.C. = 4

Signed Binary Equivalents are:
b = 0111
q = 0010

S.C. = 4
-->
R-SHIFT: 0000:0001
S.C. = 3
-->
ADD B: 0111 : 0001
R-SHIFT: 0011:1000
S.C. = 2
-->
R-SHIFT: 0001:1100
S.C. = 1
-->
R-SHIFT: 0000:1110
product is = 000001110
-----
Process exited after 48.28 seconds with return value 0
Press any key to continue . . . |

```

Figure: Booth addition and subtraction of signed 2's complement data

```
C:\Users\Acer\Documents\ca1 × + v

Kiran Joshi Sukubhattu
RESTORING DIVISION

Enter two numbers to divide
Both numbers should be less than 16

Enter the dividend: 11
Enter the divisor: 2

Expected Quotient: 5
Expected Remainder: 1

Unsigned Binary Equivalents are:
A: 01011
B: 00010
B'+1: 11110

-->
SHIFT LEFT: 00000 : 10110
-->
SUB B: 11110 : 10110
--> RESTORE
ADD B: 00000 : 10110
SHIFT LEFT: 00001 : 01100
-->
SUB B: 11111 : 01100
--> RESTORE
ADD B: 00001 : 01100
SHIFT LEFT: 00010 : 11000
-->
SUB B: 00000 : 11000
SHIFT LEFT: 00001 : 10010
-->
SUB B: 11111 : 10010
--> RESTORE
ADD B: 00001 : 10010
SHIFT LEFT: 00011 : 00100
-->
SUB B: 00001 : 00100
-----
Sign of result: 0
Remainder: 00001
Quotient: 00101
-----
Process exited after 2638 seconds with return value 0
Press any key to continue . . . |
```

Figure: Booth restoring division algorithm


```
C:\Users\Acer\Documents\ca1 × + v
Kiran Joshi Sukubhattu
          BOOTH'S MULTIPLICATION ALGORITHM
Enter two numbers to multiply:
Both must be less than 16
Enter A: 12
Enter B: 5

Expected product = 60

Binary Equivalents are:
A = 01100
B = 00101
B'+ 1 = 11011

-->
AR-SHIFT: 00000 : 00110
-->
AR-SHIFT: 00000 : 00011
-->
SUB B: 11011 : 00011
AR-SHIFT: 11101 : 10001
-->
AR-SHIFT: 11110 : 11000
-->
ADD B: 00011 : 11000
AR-SHIFT: 00001 : 11100
Product: 0000111100
-----
Process exited after 18.6 seconds with return value 0
Press any key to continue . . . |
```

Figure: Booth multiplication algorithm