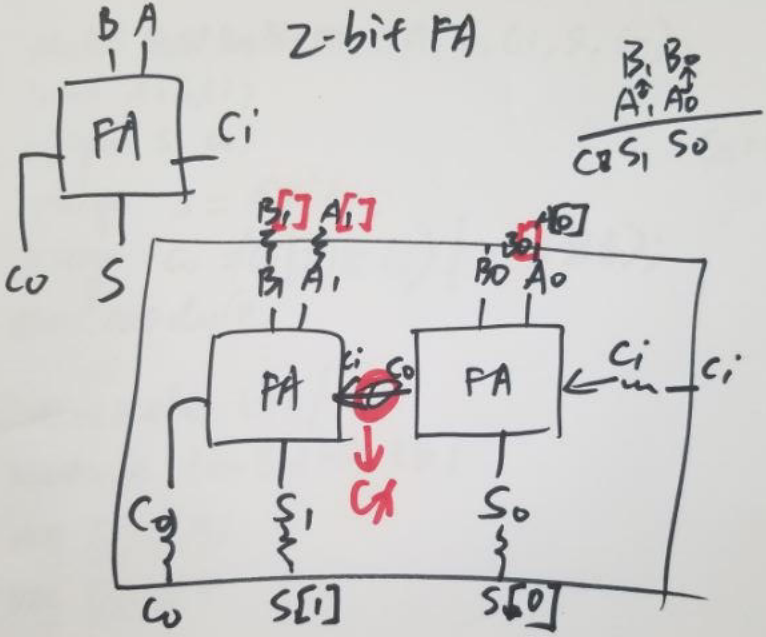
2-bit FA
B A
FA
Ci
Co S
B1 B0
A1 A0
Co S1 S0
B[1] A[1]
B[0]
B1 A1
B0 A0
FA
FA
Ci
Co
Ci
Co
S1
S0
Co
S[1]
S[0]

```
module OneBitFA (A, B, Ci, S, Co);
input A, B, Ci;
output S, Co;
assign S = A^B^Ci;
assign Co = ((A|B)& Ci) | (A&B);
endmodule

`timescale 1ns/1ps
module twoBitFA_tb;
reg [1:0] A;
reg [1:0] B;
reg Ci;
wire [1:0] S;
wire Co;
wire Cx;
```

$Co = (A+B)Ci + AB$

```
oneBitFA UUT1 (.Ci(Ci),
               .A(A[0]),
               .B(B[0]),
               .S(S[0]),
               .Co(Cx)
              );

oneBitFA UUT2 (
               .
               .
               .
               .
               .
               .
              )
```

initial begin B[1:0]=2'b0; A[1:0]=2'b0; Ci=0;

for(i=0; i<32; i=i+1)

begin

{B, A, Ci} = {B, A, Ci} + 1;

end

end

4-1 MUX