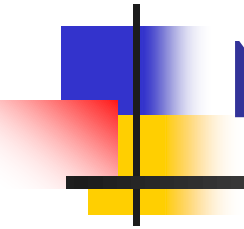
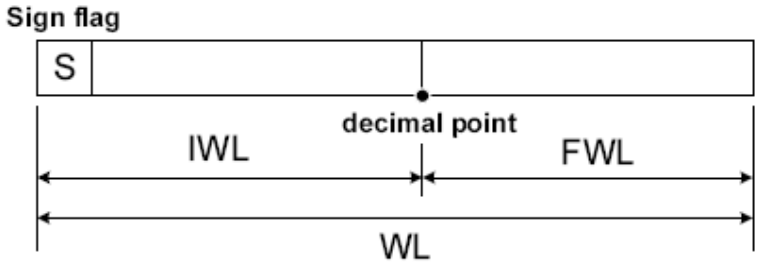


Notes for Quantization

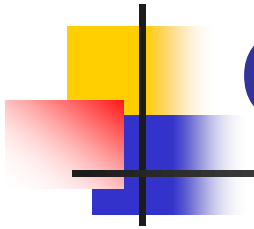
Instructor : Pei-Yun Tsai

Fixed-Point Data Format

- 3-tuple (WL, IWL, Sign)
  - WL: total wordlength
  - IWL: integer-part
  - Sign: 2's complement or unsigned



Quantization (1/2)



- Fixed-point representation
  - Truncation error
  - Round error

$3.1288_{dec}=011.001000001111\cdots_{bin}$

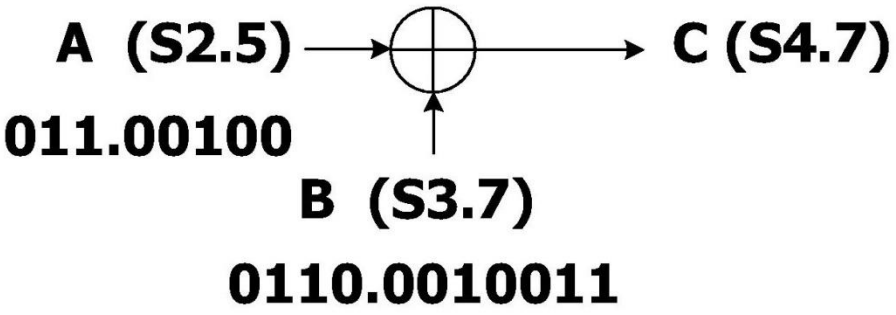
Quantization (2/2)

$X=3.1288_{dec}=011.001000001111\cdots_{bin}$

- Precision: 2^{-5}
  - (8,3,S) → S2.5
  - floor($X*2^5$)/2^5 → truncation error
- Precision: 2^{-9}
  - (12,3,S) → S2.9
  - Round($X*2^9$)/2^9 → round error

Fixed-point Addition (1/2)

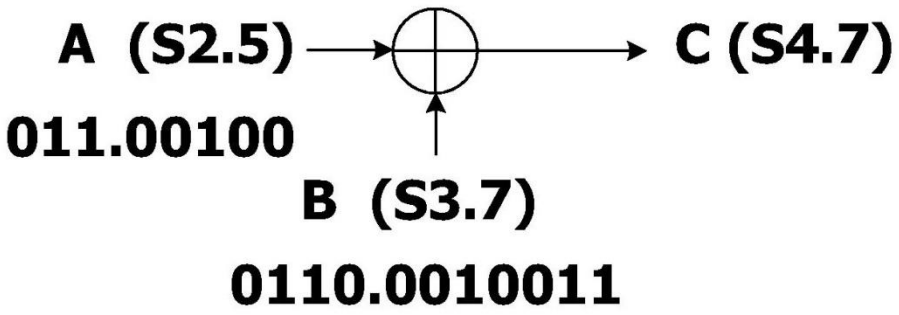
- Word-length of the integer part is increased.
- Word-length of the fractional part follows the more precise one.




```
  0011.0010000
+ 0110.0010011
-----------------
  01001.0100011
```

Fixed-point Addition (2/2)

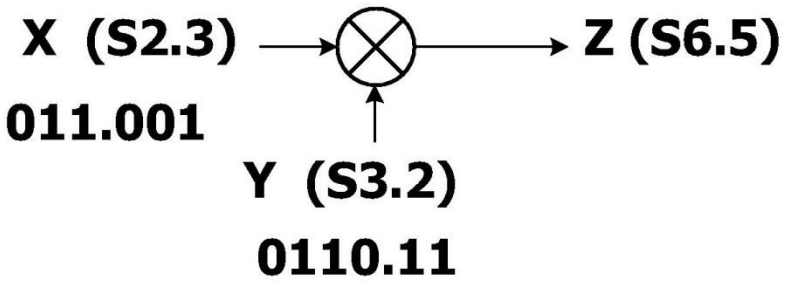
```
wire [7:0] A;
wire [10:0] B;
reg [11:0] C;

always @(A or B)
begin
    C={{2{A[7]}},A,2'b00}+{B[10],B};
end
```




```
  0011.0010000
+ 0110.0010011
-------------
  01001.0100011
```

Fixed-point Multiplication (1/3)

- Word-length of the integer part may be increased.
- Word-length of the fractional part increases severely. Hence, truncation is necessary.



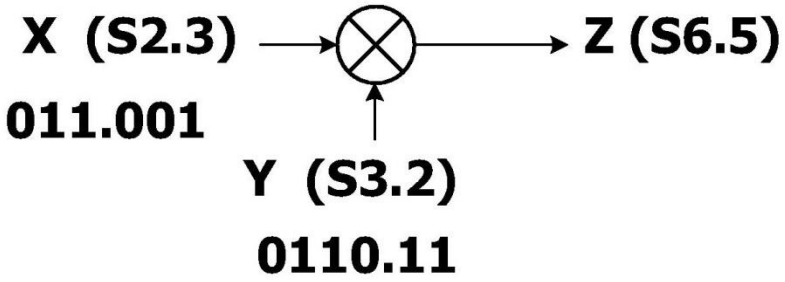

```
      011.001
  ×)  0110.11
  ------------
      011001
     011001
    011001
   011001
  ------------
  0010101.00011
```

Fixed-point Multiplication (2/3)

- Assume T(Z) is the representation of Z after truncation of several LSBs of Z.
  - Z=XY
  - $T(Z)= \text{floor}(Z*2^a)/2^a$
- Criterion
  - max(|T(Z)-Z|) < upper bound
  - avg($|T(Z)-Z|^2$) < upper bound

Fixed-point Multiplication (3/3)

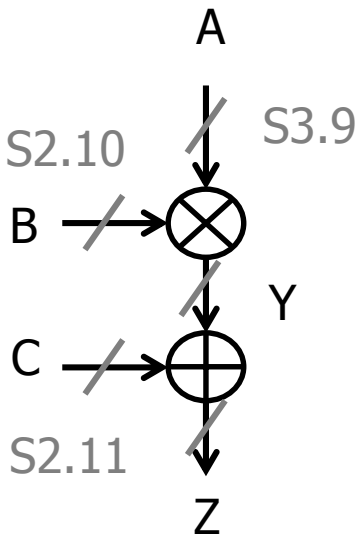
```
wire signed [5:0] X;
wire signed [5:0] Y;
wire signed [11:0] Z;
wire signed [8:0] W;

assign Z=X*Y;
assign W=Z[11:3];
```



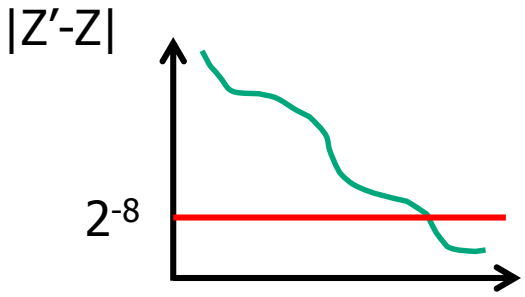

```
      011.001
  ×)  0110.11
-------------
     011001
    011001
   011001
  011001
-------------
0010101.00011
```

Example

$Z=AB+C=Y+C$ **Criterion**: $|Q(Z)-Z|<2^{-7}$

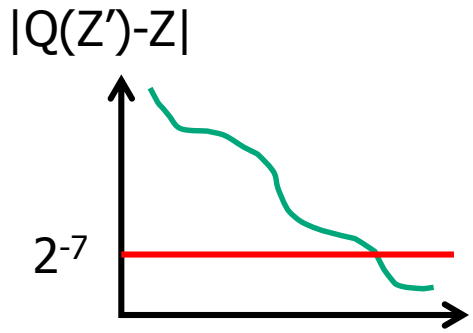


1) $Z'=Q(Y)+C$



Fractional Word-length of Y

2) $Q(Z')-Z$



Fractional Word-length of Z'

Verilog File Output

```
MAC u1(.InA(A), .InB(B), .OutC(Z), .OutD(Y), .RST(RST), .CLK(CLK));

initial
begin
            mcd1 = $fopen("xyz.txt"); CLK=1'b0;
        RST=1'b0; A=10'd700; B=12'd603;
  #10  RST=1'b1; A=10'd602; B=12'd622;
  #10  RST=1'b1; A=10'd23;   B=12'd908;
  #10  RST=1'b1; A=10'd1022; B=12'd305;
  #10   RST=1'b1; A=10'd211;  B=-12'd768;
  #10   RST=1'b1; A=10'd99;   B=-12'd999;
  #10   RST=1'b1; A=10'd505;  B=-12
  #10 $fclose(mcd1);
  #10  $finish;
end
```

```
always @ (posedge clk)
begin
$fwrite(mcd1,"%d %d \n",Z,Y);
end

always
begin
#5 CLK=~CLK;
end
```