1

3840 x 2160 x 60fps x 16bit ÷ 8bit

= 995,328,000 B/S

≒ 995 MB/S

	Inpot B/W	outpot B/W
TSD		10 MB/S
Mpeg	10MB/S	995MB/S
de-Interaced	995MB/S	995 MB/S
scaler	995MB/S	995MB/S
Mixer	995MB/S	995MB/S
GA	X	X
CPU	100MB/S	
출력		995MB/S

다 더하면 8080 MB/S

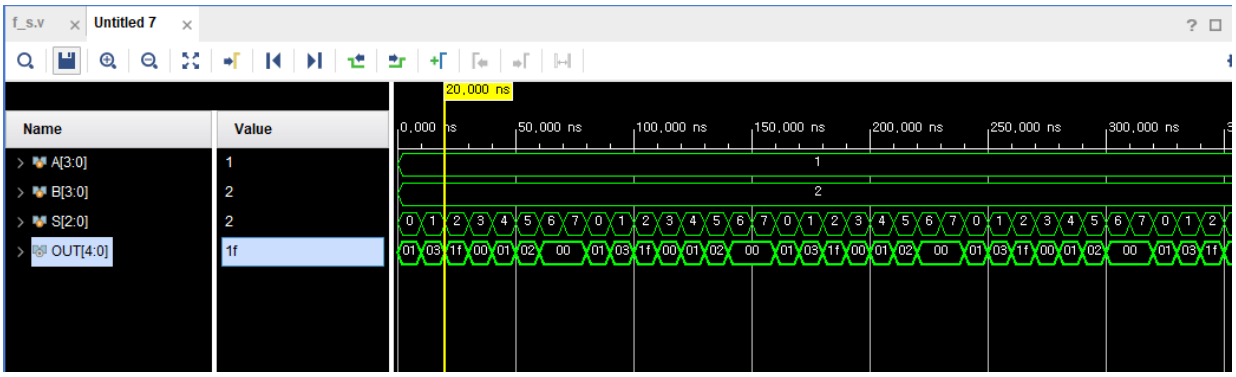
128 bit 버스 이므로 ⇒ 8080 ÷ 128 X 8 = 505 Mbps

⇒ 505MHz

2.

```
module ff(out, a, b, s);
input [3:0] a, b;
input [2:0] s;
output reg [4:0] out;

always @(a, b, s)
begin
case (s)
	3'b000: out = a;
	3'b001: out = a+b;
	3'b010: out = a-b;
	3'b011: out = a/b;
	3'b100: out = a%b;
	3'b101: out = a<<1;
	3'b110: out = a>>1;
	3'b111: out = (a>b);
	default: $display("Invalid");
endcase
end

endmodule

module stimulus;
reg [3:0] A, B;
reg [2:0] S;
wire [4:0] OUT;

ff example(OUT, A, B, S);

initial
begin
	A=4'b0001; B=4'b0010;
	S=3'b000;
end
```

```
always
    #10 S=S+1'b1;


endmodule
```



3.

```
module example( z, x, clock, clear);

output z;
reg z;
input x;
input clock, clear;

parameter S0 = 3'd0 ,
          S1 = 3'd1 ,
          S2 = 3'd2 ,
          S3 = 3'd3 ,
          S7 = 3'd7;
reg [2: 0] state;
reg [2: 0] next_state;

always @(posedge clock)
        if( clear )
                state <= S0;
        else
                state <= next_state;

always @(state)
begin
        z = 1'd0;
        case(state)
                S0 : ;
                S1 : ;
                S2 : ;
                S3 : ;
                S7 : z = 1'd1;
        endcase
end
```

```
always @(state or x)   // next state 를 결정하는 로직
begin
        case(state)
                S0 : if(x)
                        next_state = S1;
                     else
                        next_state = S0;
                S1 : if(x)
                        next_state = S3;
                     else
                        next_state = S2;
                S2 : if(x)
                        next_state = S0;
                     else
                        next_state = S7;
                S3 : if(x)
                        next_state = S2;
                     else
                        next_state = S7;
                S7 :
                        next_state = S0;

        endcase
end
```