Mantıksal Sistem Tasarımı – BLM 201

Hafta 7: Veriyolu Elemanları
Bölüm II

Fenerbahçe Üniversitesi

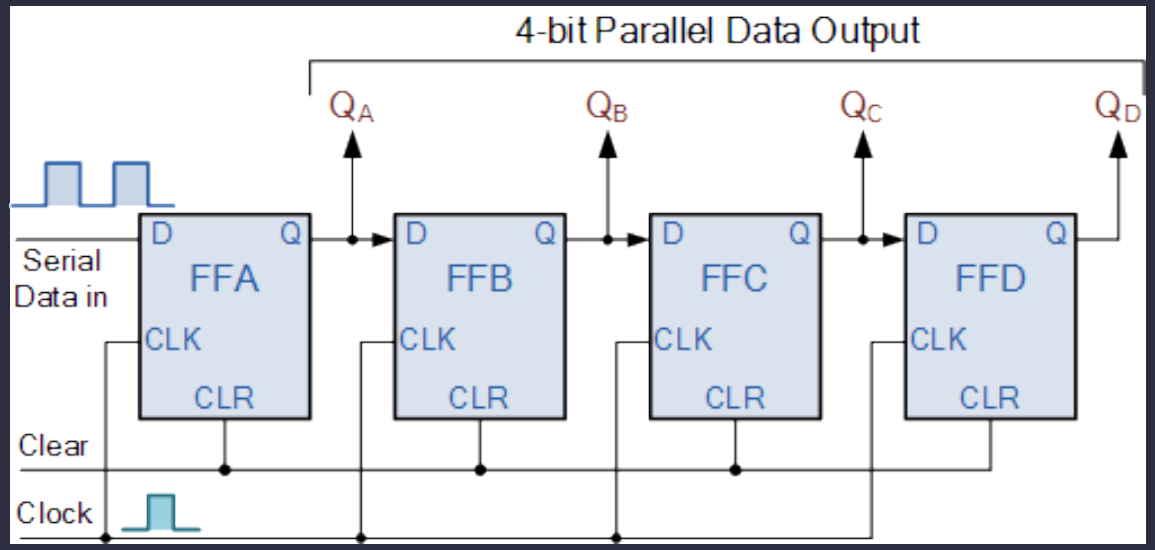
7. Hafta İçeriği

- Veriyolu Elemanları
  - Veriyolu Elemanları Verilog Gösterimleri

Shift Register

Veriyolu Elemanları

Seri Giriş

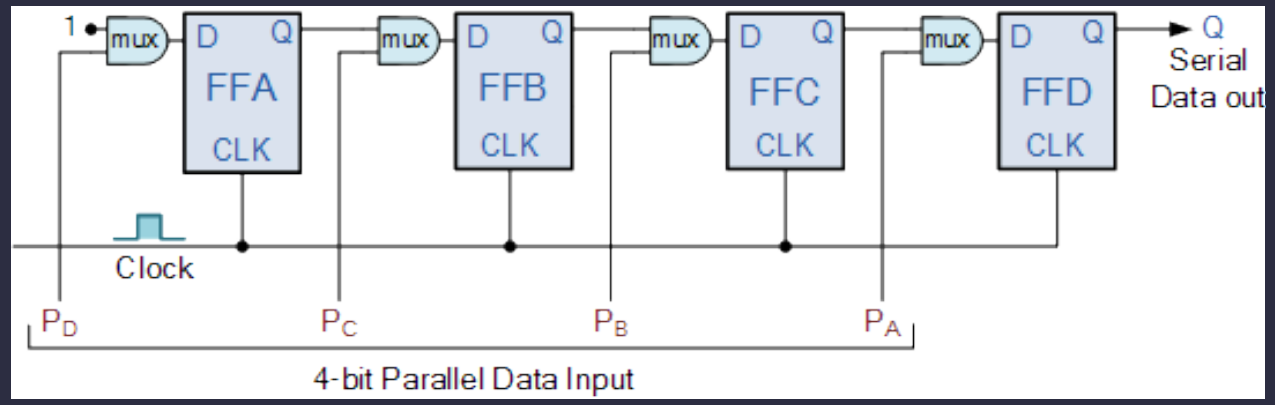


Verilog Tasarım

```
module myModule(input clk, input  in, input rst, output reg [3:0] out);

        reg [3:0] outNext;

        always@(posedge clk) begin
                out <= outNext;
        end

        always@(*) begin
                outNext = {out[2:0], in};
                if(rst)
                        outNext = 0;
        end

endmodule
```

Veriyolu Elemanları

Paralel Giriş



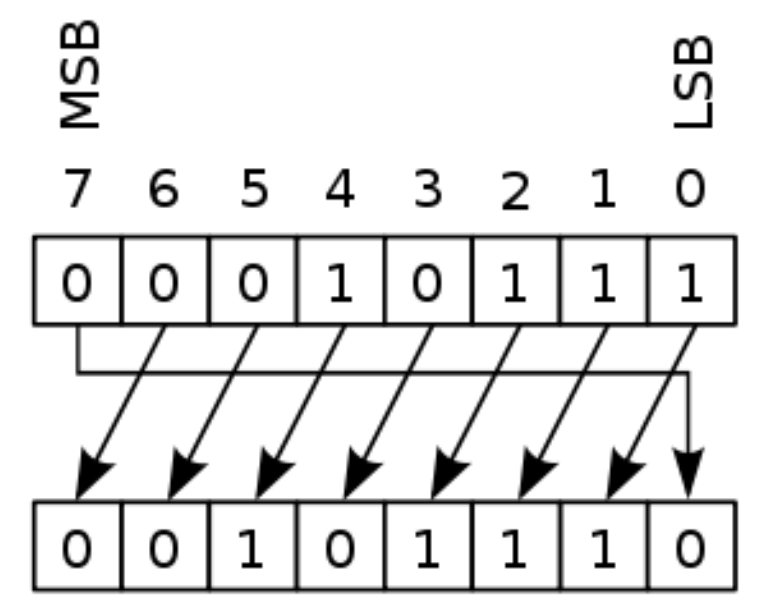
Verilog Tasarım

```
module myModule(input clk, input [3:0] in, input yukle, input output reg
[3:0] out);

        reg [3:0] outNext;

        always@(posedge clk) begin
                out <= outNext;
        end

        always@(*) begin
                outNext = {out[2:0], 1'b1};
                if(yukle)
                        outNext = in;
        end

endmodule
```

Rotate Register

Veriyolu Elemanları

Rotate Register



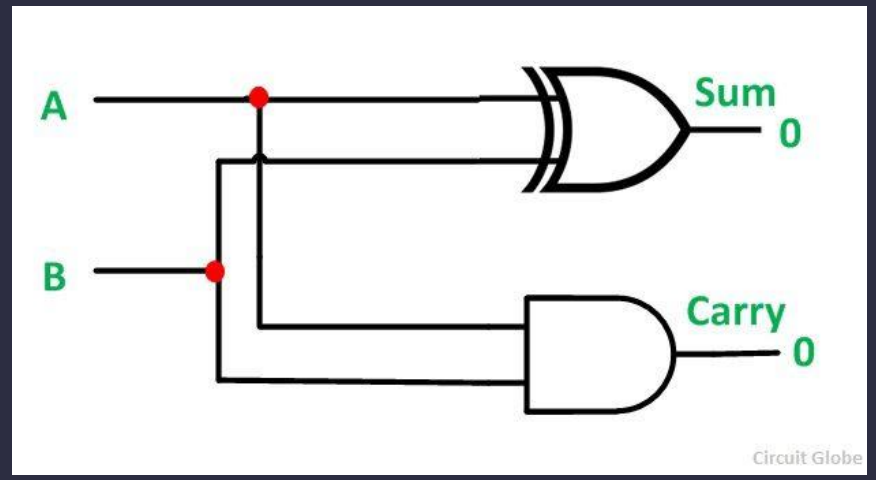
Verilog Tasarım

```
module myModule(input clk, output reg [7:0] rotateReg);

        reg [7:0] rotateRegNext;

        always@(posedge clk) begin
                rotateReg <= rotateRegNext;
        end

        always@(*) begin
                rotateRegNext = {rotateReg[6:0],rotateReg[7]};
        end

endmodule
```

Adder

Veriyolu Elemanları

Adder

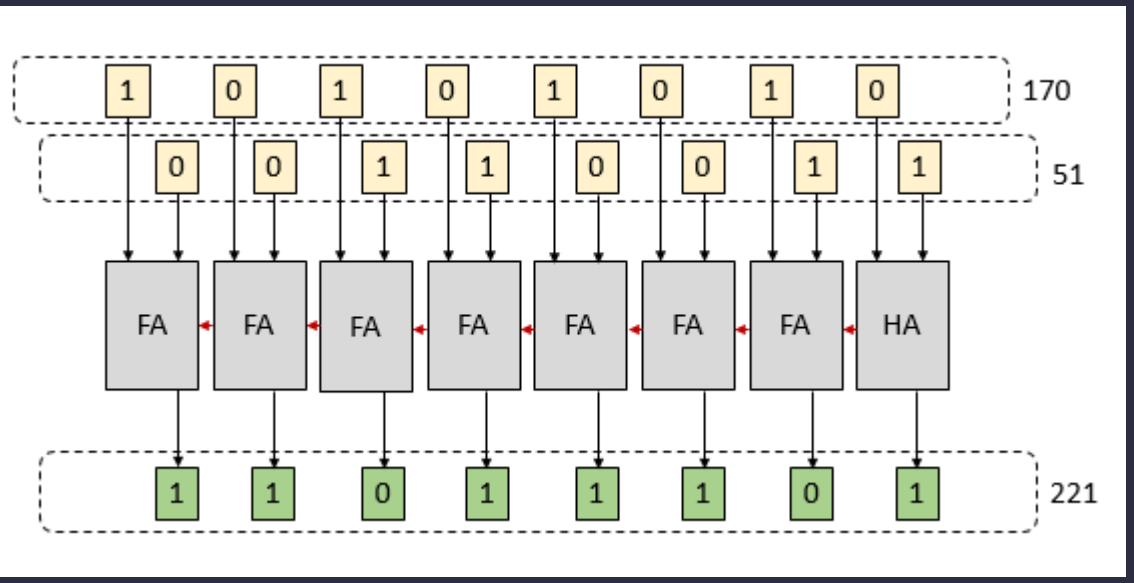


Verilog Tasarım

```
module myModule(input in1, input in2, output reg sum, output reg
carry);

        always@(*) begin
                sum = in1 ^ in2;
                carry = in1 & in2;
        end

endmodule
```

Veriyolu Elemanları

Adder



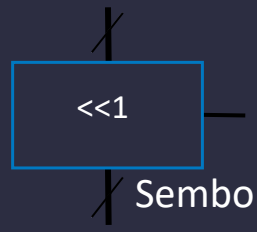
Verilog Tasarım

```
module myModule(input [7:0] in1, input [7:0] in2, output reg [7:0] sum,
output reg carry);

        always@(*) begin
                {carry, sum} = in1 + in2;
        end

endmodule
```

Shifter

Veriyolu Elemanları

Shifter



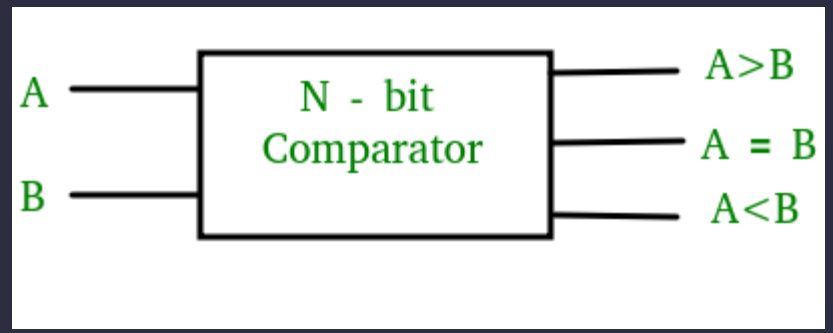
Verilog Tasarım

```
module myModule(input [7:0] in, output reg [7:0] out);

        always@(*) begin
                out = in << 1;
        end

endmodule
```

Karşılaştırıcı

Veriyolu Elemanları

Comparator



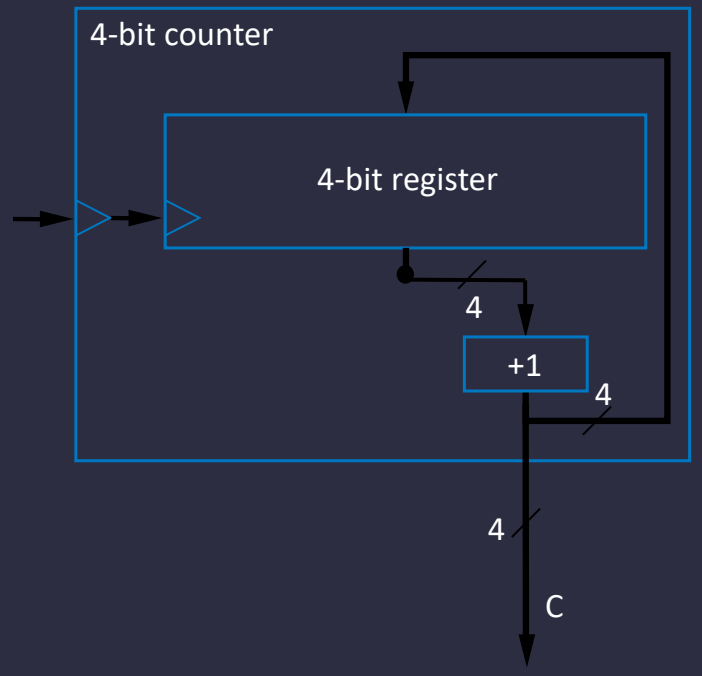
Verilog Tasarım

```
module myModule(input  [7:0] a, input [7:0] b, output reg bigger,
output reg equal, output reg less);

        always@(*) begin
                bigger = a > b;
                equal = a == b;
                less = a < b;
        end

endmodule
```

Counter

Veriyolu Elemanları

Counter



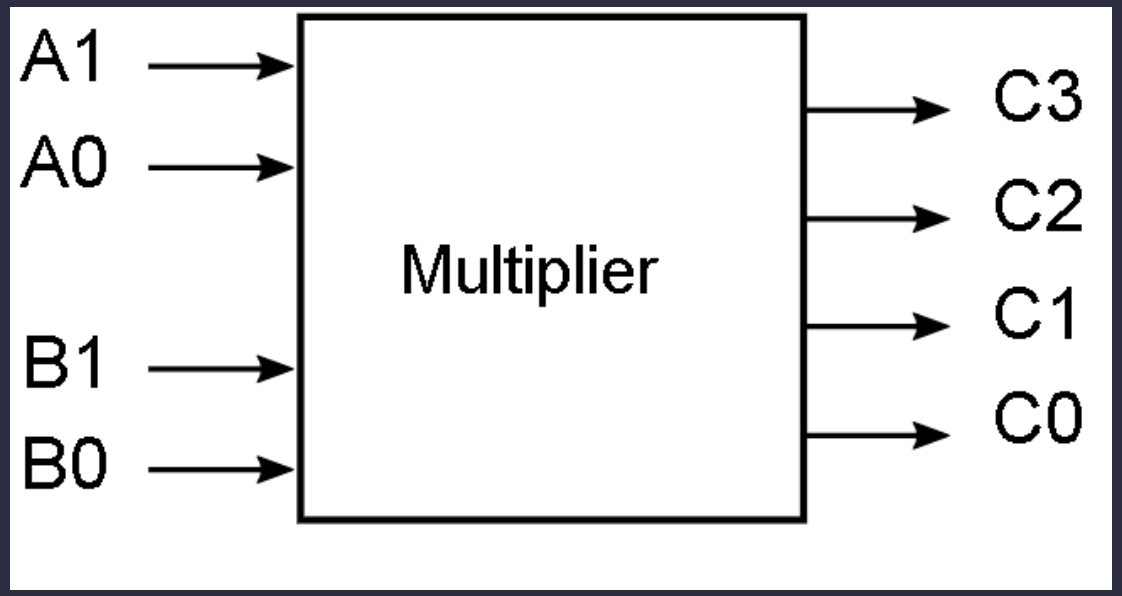
Verilog Tasarım

```
module myModule(input clk, output reg [3:0] counter);

        reg [3:0] counterNext;

        always@(posedge clk) begin
                counter <= counterNext;
        end

        always@(*) begin
                counterNext = counter + 1;
        end

endmodule
```

Çarpıcı

Veriyolu Elemanları

Çarpıcı



Verilog Tasarım

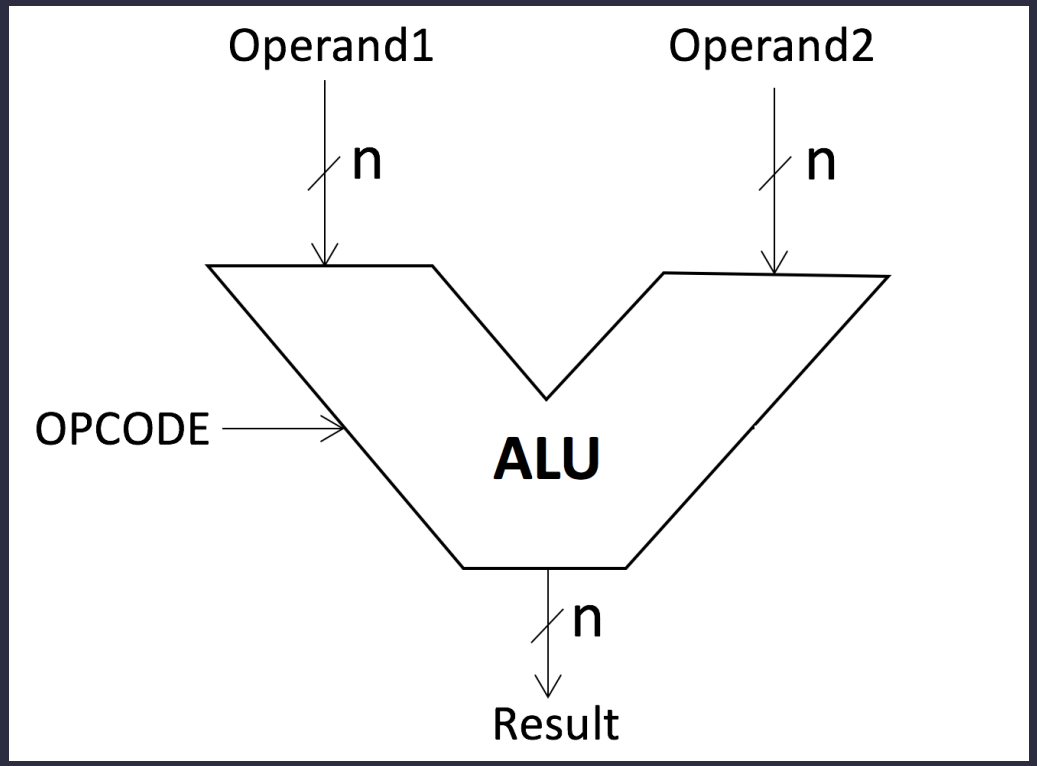
```
module myModule(input [1:0] a, input [1:0] b, output [3:0] c);

        always@(*) begin
                c = a * b;
        end

endmodule
```

ALU

Veriyolu Elemanları

ALU

- Toplama
- Çıkarma
- AND
- OR
- XOR
- NOT
- Sağa kaydırma
- Sola kaydırma



Veriyolu Elemanları

ALU

Verilog Tasarım

```
module myModule(input [7:0] operand1, input [7:0] operand2, input [2:0] opCode ,output reg [15:0] result);

        always@(*) begin
                if(opCode == 0)
                        result = operand1 + operand2;
                else if(opCode == 1)
                        result = operand1 - operand2;
                else if(opCode == 2)
                        result = operand1 & operand2;
                else if(opCode == 3)
                        result = operand1 | operand2;
                else if(opCode == 4)
                        result = operand1 ^ operand2;
                else if(opCode == 5)
                        result = ~operand1;
                else if(opCode == 6)
                        result = operand1 << 1;
                else if(opCode == 7)
                        result = operand1 >> 1;
        end

endmodule
```