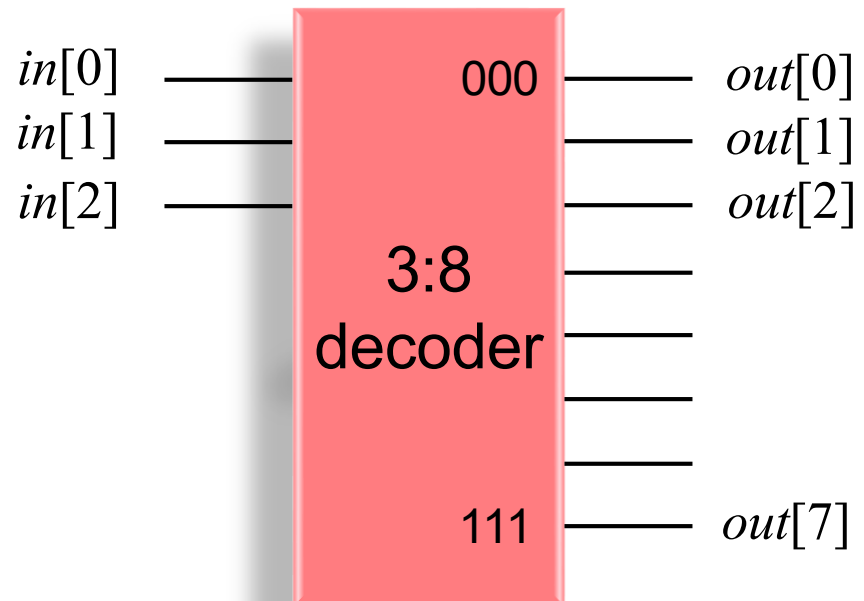


Дешифратор және Шифратор.
Комбинациялық логикалық құрылғы
жобалау және синтездеу.

Дешифратор

- i кірістері бар декодердің 2^i шығысы болады.
- Қолдану аясы:
 - Жад ішіндегі сөзді таңдау.
 - Көптеген модульдер қосылған кезде шинаға қосылған бір модульді таңдау (мысалы, үш күйдегі драйверлер).



3:8 Дешифратор

3 to 8 Line Decoder

Inputs			Outputs							
x	y	z	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

$$D_0(x, y, z) = \bar{x} \bar{y} \bar{z}$$

$$D_1(x, y, z) = \bar{x} \bar{y} z$$

$$D_2(x, y, z) = \bar{x} y \bar{z}$$

$$D_3(x, y, z) = \bar{x} y z$$

$$D_4(x, y, z) = x \bar{y} \bar{z}$$

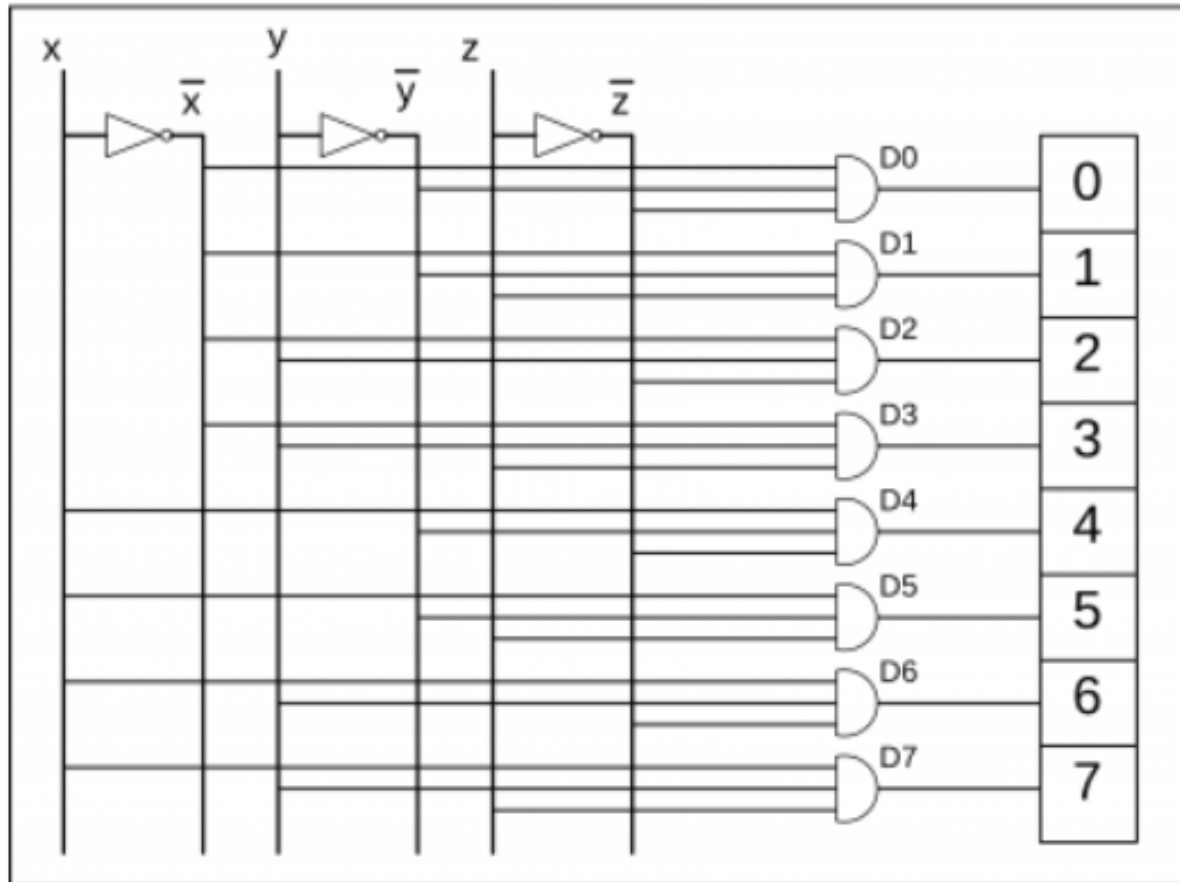
$$D_5(x, y, z) = x \bar{y} z$$

$$D_6(x, y, z) = x y \bar{z}$$

$$D_7(x, y, z) = x y z$$

3:8 Дешифратор

3 x 8 line Decoder Logic Diagram



3:8 Дешифратор

Individual wires

- 8 шығысы бар блок шын мәніне сәйкес 8 түрлі дизайн болып табылады.

```
// in[2:0] may be a wire, reg, or input
wire [7:0] out1;

// Individual wires
assign out1[0] = ~in[2] & ~in[1] & ~in[0];
assign out1[1] = ~in[2] & ~in[1] & in[0];
assign out1[2] = ~in[2] & in[1] & ~in[0];
assign out1[3] = ~in[2] & in[1] & in[0];
assign out1[4] = in[2] & ~in[1] & ~in[0];
assign out1[5] = in[2] & ~in[1] & in[0];
assign out1[6] = in[2] & in[1] & ~in[0];
assign out1[7] = in[2] & in[1] & in[0];
```

3:8 Дешифратор (case Statement).

- 8 биттік шина негізінде case операторының көмегімен жиналған Дешифратор.

```
reg [7:0] out2;  
  
// Example 1: case statement  
always @(in) begin  
    case(in)  
  
        3'b000: begin    out2=8'b00000001;    end  
        3'b001: begin    out2=8'b00000010;    end  
        3'b010: begin    out2=8'b00000100;    end  
        3'b011: begin    out2=8'b00001000;    end  
        3'b100: begin    out2=8'b00010000;    end  
        3'b101: begin    out2=8'b00100000;    end  
        3'b110: begin    out2=8'b01000000;    end  
        3'b111: begin    out2=8'b10000000;    end  
  
    endcase  
end
```

3:8 Дешифратор (case Statement).

- Үш тәуелсіз кірісі бар қарапайым жағдай.

```
inputs a, b, c
reg [7:0] out2;

always @(a or b or c) begin
    case({a,b,c})

        3'b000: begin    out2=8'b00000001;    end
        3'b001: begin    out2=8'b00000010;    end
        3'b010: begin    out2=8'b00000100;    end
        3'b011: begin    out2=8'b00001000;    end
        3'b100: begin    out2=8'b00010000;    end
        3'b101: begin    out2=8'b00100000;    end
        3'b110: begin    out2=8'b01000000;    end
        3'b111: begin    out2=8'b10000000;    end

    endcase
end
```

Дешифраторды тексеру ортасы.

- Бұл сынақ модулінің ішіндегі тәуелсіз блоктар.

```
//----- Main test loop
// This block is executed once at the
// beginning of the simulation.
initial begin
    $write("Simulation beginning\n");
    #100;
    in = 3'b000; #100;
    in = 3'b001; #100;
    in = 3'b010; #100;
    in = 3'b011; #100;
    in = 3'b100; #100;
    in = 3'b101; #100;
    in = 3'b110; #100;
    in = 3'b111; #100;

    $stop;          // ends simulation
end
```

```
//----- Print statements
always @(in) begin
    #10; // gives a tiny delay after "in" changes
    $write("in = %b, out1 = %b\n", in, out1);
end
```

```
reg      [2:0] in;
reg      [7:0] out1;

// example decoder hardware
always @(in) begin
    case(in)
        3'b000: begin out1=8'b00000001; end
        ...
    endcase
end
```

Simulation beginning

in = 000,	out1 = 00000001
in = 001,	out1 = 00000010
in = 010,	out1 = 00000100
in = 011,	out1 = 00001000
in = 100,	out1 = 00010000
in = 101,	out1 = 00100000
in = 110,	out1 = 01000000
in = 111,	out1 = 10000000

3:8 Дешифратор.

Солға жылжыту операторы

- 8 шығысты бір регистр шинасы ретінде қарастырыңыз.

```
// in[2:0] may be a wire, reg, or input  
reg [7:0] out3;
```

```
// Example 3: reg with <<  
always @(in) begin  
    out3 = 8'b0000_0001 << in;  
end
```

3:8 Дешифратор. Солға жылжыту операторы.

- 8 шығысты бір сымды шина ретінде қарастырыңыз.

```
// in[2:0] may be a wire, reg, or input  
wire [7:0] out4;
```

```
// Example 4: wire with <<  
assign out4 = 8'b0000_0001 << in;
```

3:8 Дешифратор, Жартылай анықталған шығыс жағдайлары;

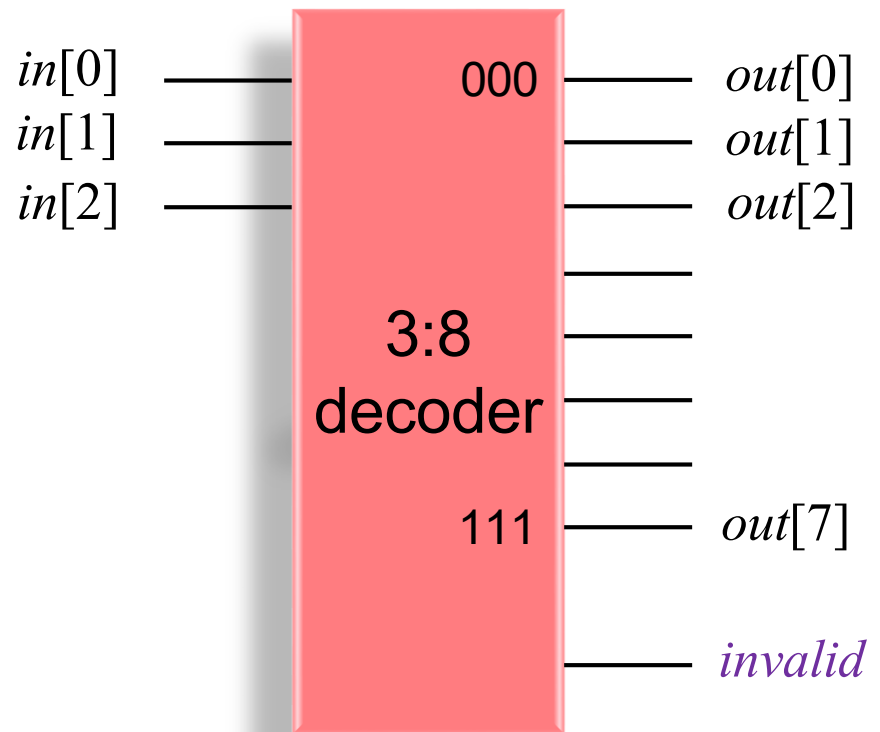
- Кіріс комбинацияларының тек 4-і ғана анықталған: 000, 001, 100, 110.
- Сондай-ақ жүзеге асыра алады:
 - 1) әрқашан блоктың басында бастапқы параметрді орнату.
 - 2) бастапқы параметрді орнатыңыз, содан кейін «if» мәлімдемелерін пайдаланыңыз.
 - 3) көптеген басқа опциялар.

```
reg [7:0] out5;

// Example 5: partially-defined outputs
always @(in) begin
    case(in)
        3'b000: begin    out5=8'b00000001;    end
        3'b001: begin    out5=8'b00000010;    end
        //
        //
        3'b100: begin    out5=8'b00010000;    end
        //
        3'b110: begin    out5=8'b01000000;    end
        //
        default:begin    out5=8'bxxxxxxxx;    end
    endcase
end
```

Қосымша жарамсыз Шығыс сигналы бар Дешифратор.

- Келесі мысал бірдей жарамды кіріс комбинациялары бар 3:8 декодеріне арналған: 000, 001, 100, 110.
- Бірақ бұл мысалда кірістің жарамды мәні болмаған кезде:
 - Сегіз негізгі шығыс барлық нөлдер болуы керек.
 - Жаңа жарамсыз шығыс сигналы жоғары орнатылған.



3:8 Дешифратор, *invalid* Output.

- Алдыңғы мысалмен бірдей, тек:

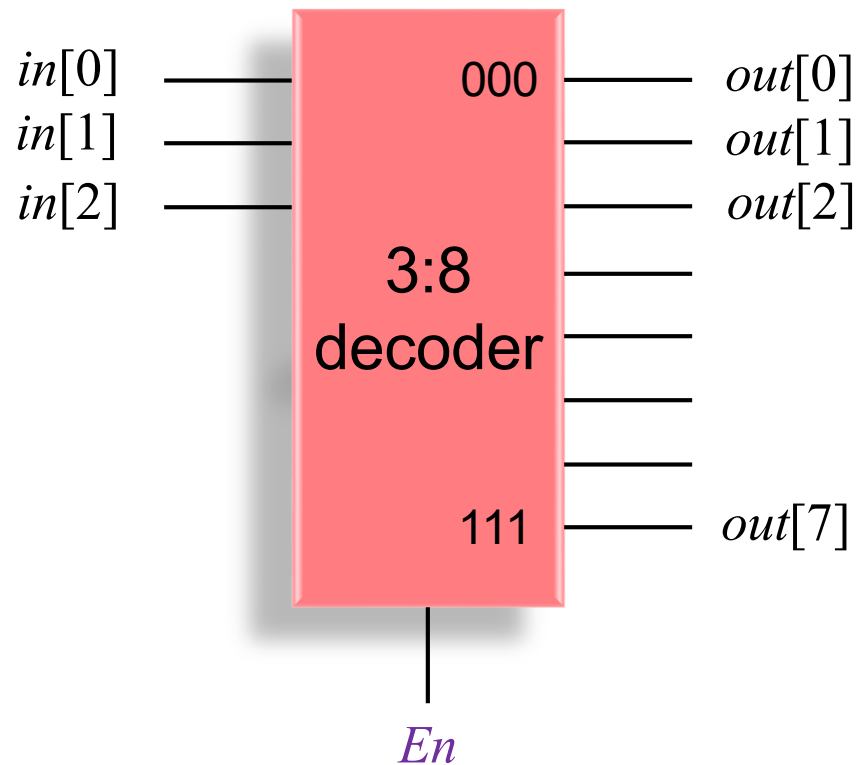
- Анықталмаған жағдайлар нөлдік нәтижеге ие.
- кіріс шығыстары анықталмаған мәнге тең болғанда жарамсыз сигнал орнатылады.

```
reg [7:0] out6;

// Example 6: invalid output signal
always @(in) begin
    invalid = 1'b0;
    case(in)
        3'b000: begin    out6=8'b00000001;    end
        3'b001: begin    out6=8'b00000010;    end
        //
        //
        3'b100: begin    out6=8'b00010000;    end
        //
        3'b110: begin    out6=8'b01000000;    end
        //
        default: begin
            out6      = 8'b0000_0000;
            invalid = 1'b1;
        end
    endcase
end
```

Дешифратор Enabled шығысымен

- Декодерлерде шығыстардың жалпы күйін басқаратын Enable кірісі жиі болады.
 - En кірісі жоғары болғанда, ол қалыпты декодер ретінде жұмыс істейді.
 - En кірісі төмен болғанда, барлық шығыстар төмен болады.



3:8 Дешифратор, Enabled ШЫҒЫСЫМЕН

- 7-мысал:
Қосылған шығыс:
- En кірісі жоғары болғанда, ол қалыпты декодер ретінде жұмыс істейді.
- En кірісі төмен болғанда, барлық шығыстар төмен болады.

```
reg [7:0] out7;

// Example 7: Enabled output version 1
always @(in) begin
    // enabled, function like a normal encoder
    if (En == 1'b1) begin
        case(in)
            3'b000: begin out7=8'b00000001; end
            3'b001: begin out7=8'b00000010; end
            3'b010: begin out7=8'b00000100; end
            3'b011: begin out7=8'b00001000; end
            3'b100: begin out7=8'b00010000; end
            3'b101: begin out7=8'b00100000; end
            3'b110: begin out7=8'b01000000; end
            3'b111: begin out7=8'b10000000; end
        endcase
    end
    // not enabled, set all outputs to zero
    else begin
        out7=8'b00000000;
    end
end
```

3:8 Дешифратор, Enabled ШЫҒЫСЫМЕН

- 8-мысал:
Қосылған шығыс:
- En кірісі жоғары болғанда, ол қалыпты декодер ретінде жұмыс істейді.
- En кірісі төмен болғанда, барлық шығыстар төмен болады.
- Көптеген іске асыру мүмкін.

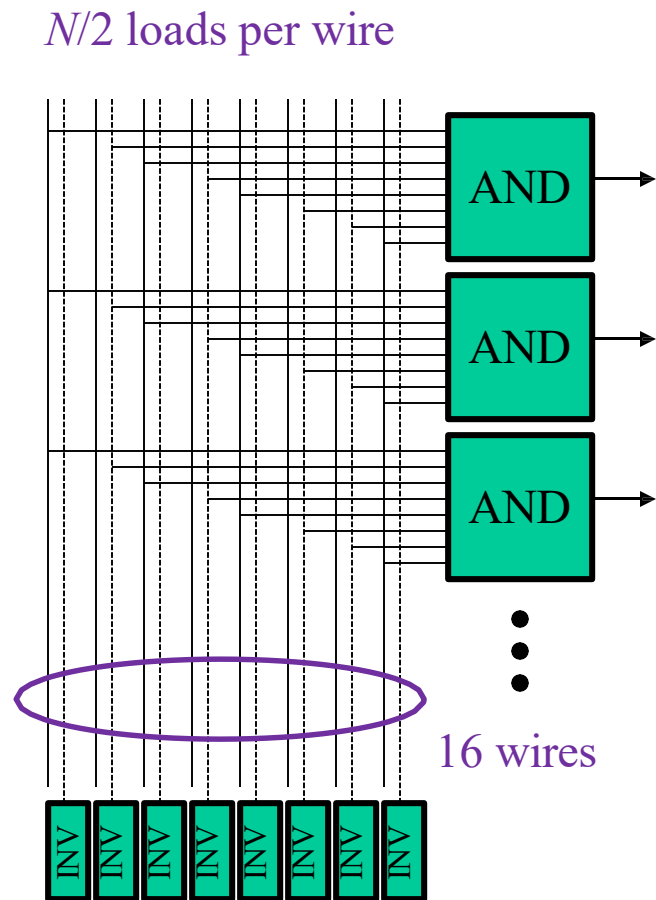
```
reg [7:0] out8;

// Example 8: Enabled output version 2
always @(in) begin
    // default. Initialize output as if not enabled.
    out8=8'b00000000;

    // if enabled, function like a normal encoder
    if (En == 1'b1) begin
        case(in)
            3'b000: begin out8=8'b00000001; end
            3'b001: begin out8=8'b00000010; end
            3'b010: begin out8=8'b00000100; end
            3'b011: begin out8=8'b00001000; end
            3'b100: begin out8=8'b00010000; end
            3'b101: begin out8=8'b00100000; end
            3'b110: begin out8=8'b01000000; end
            3'b111: begin out8=8'b10000000; end
        endcase
    end
end
```

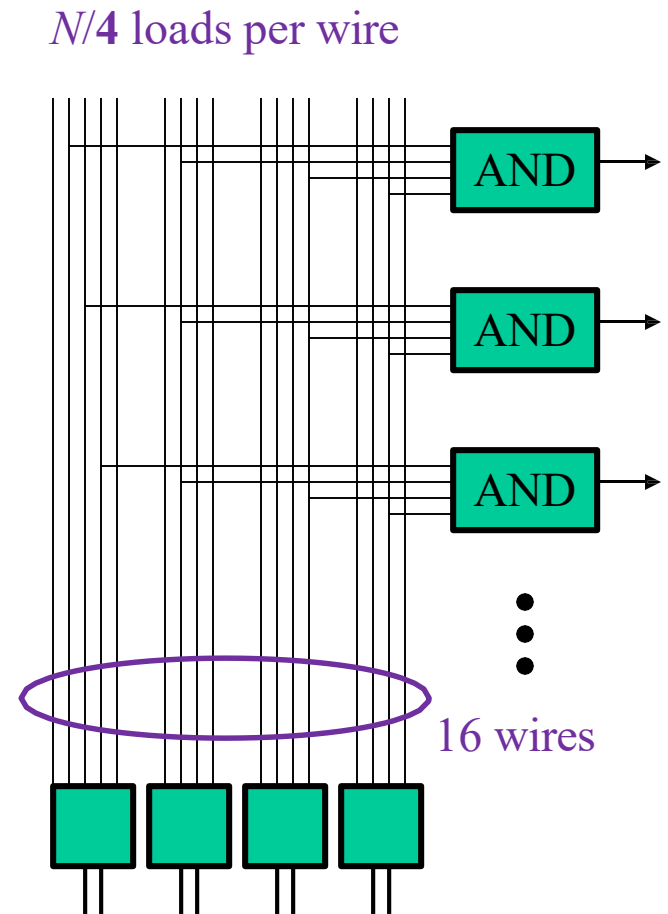
«Қолмен» жасалған Дешифратор Жад массивінің дизайны.

- Мысалы: 256-шығыс (8 кіріс бит)
- Тікелей тәсіл:
- 8 кіріс сымын және төңкерілген нұсқаларды (16 сым) массив бойымен бағыттаңыз.
- Әрбір сөз 8 кірісті ЖӘНЕ қақпасын пайдаланады.
- Әрбір ұзын сымның $N/2=128$ жүктемесі бар.



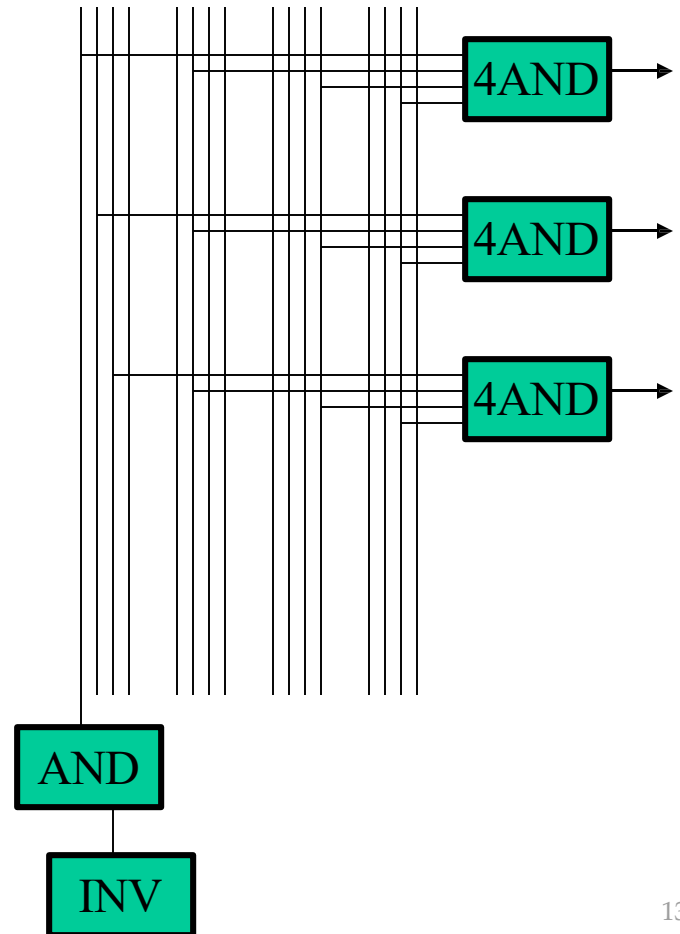
Алдын ала кодтауды қолдану арқылы Дешифратор дизайны.

- Мысал: алдын ала кодпен 256-шығыс (8 кіріс бит).
 - Массив бойындағы сымдар: $22 * (8/2) = 4 = 16$ -дан тұратын 4 топ (алдын ала кодталмаған сияқты).
 - Әрбір шығыс 4 кірісті ЖӘНЕ қақпаны пайдаланады (тезірек).
 - Әрбір ұзын сымның $N/4=64$ қақпа жүктемесі бар (басқа тәсілдің жартысы!).
 - Алдын ала кодтау үлкен декодерлермен жақсы жұмыс істейді.



Decoder Design Using Predecoding

- Critical path
 - Predecoder + 4-input AND
 - Neglect assertion level because we'll have to add a number of inverters (buffers) anyway
 - INV + 2-input AND + 4-input AND
 - May be able to use NANDs instead of ANDs



Алдын ала кодтауды қолдану арқылы декодер дизайны.

- Мысал: 3 адрес биттерінің алдын ала кодтау топтары.

Алдын ала кодтау топтары: бірдей 8:256 декодер үшін $3 + 3 + 2$.

Әрбір 3 кіріс алдын ала кодтау тобында $2^3 = 8$ шығыс сымы бар.

Әрбір 3 кірісті алдын ала кодталған сымның $N/8$ жүктемесі бар.

Барлығы $8 + 8 + 4 = 20$ алдын ала кодталған сымдар.

- Мысал: 4 адрес биттерінің алдын ала кодтау топтары.

Алдын ала кодтау топтары: бірдей 8:256 декодер үшін $4 + 4$.

Әрбір алдын ала кодтау тобында $2^4 = 16$ шығыс сымы бар.

Әрбір алдын ала кодталған сымның $N/16$ жүктемесі бар.

Барлығы $16 + 16 = 32$ алдын ала кодталған сымдар.