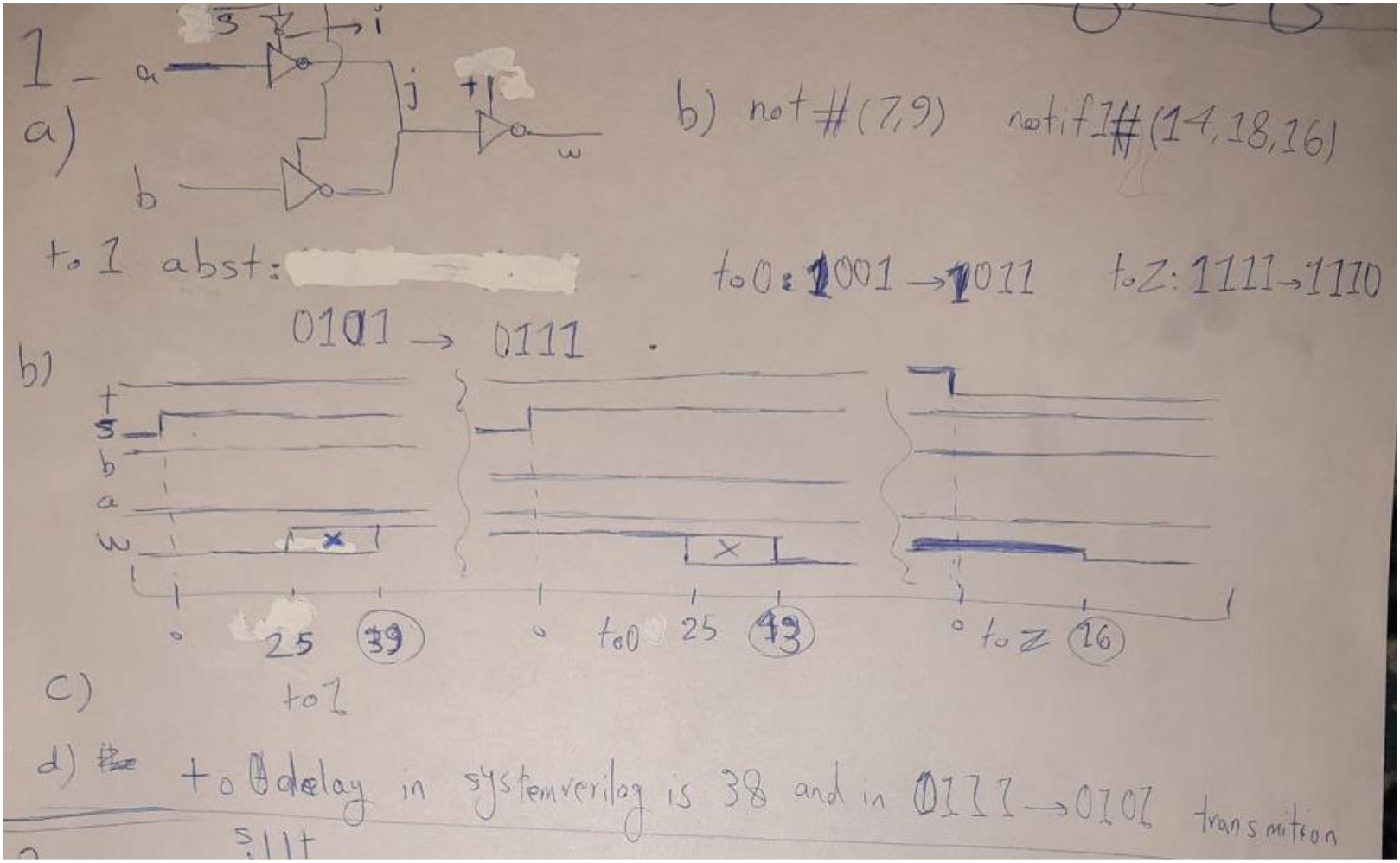
علی عطاللهی

CA2

810199461

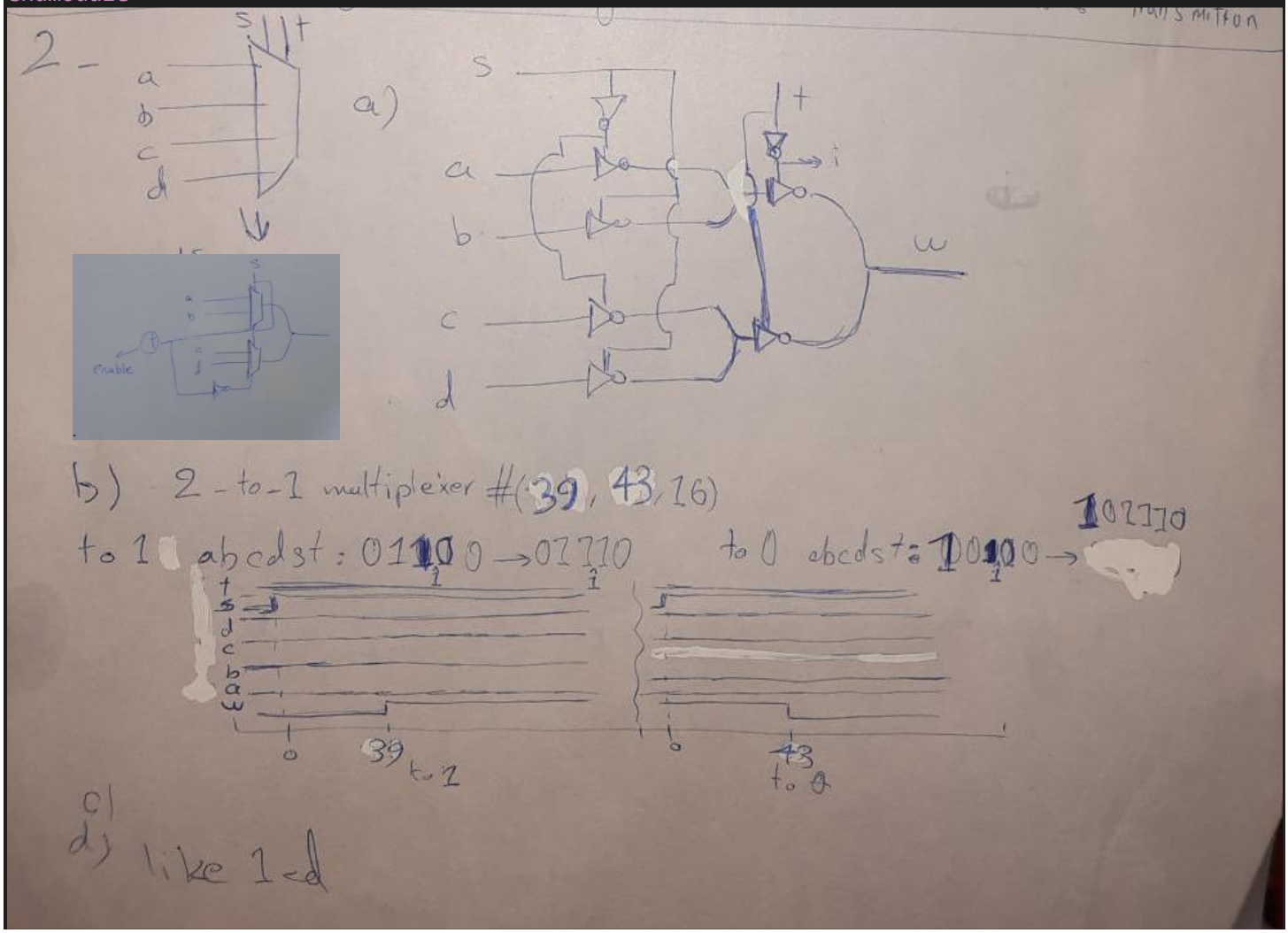



```
`timescale 1ns/1ns
module not_ (input c,output w2);
    supply1 Vdd;
    supply0 Gnd;
    pmos #(4,7,9) T5(w2,Vdd,c);
    nmos #(3,5,7) T6(w2,Gnd,c);
endmodule
module not_ctrl_ (input c,EN1,output w1);
    wire i1,i2,i3;
    supply1 Vdd;
    supply0 Gnd;
    pmos #(4,7,9) T1(i2,Vdd,c);
    pmos #(4,7,9) T2(w1,i2,i1);
    pmos #(4,7,9) T3(i1,Vdd,EN1);
    nmos #(3,5,7) T4(i1,Gnd,EN1);
    nmos #(3,5,7) T5(w1,i3,EN1);
    nmos #(3,5,7) T6(i3,Gnd,c);
endmodule
```

```
module CA2_E1 (input a,b,s,t,output w);
    wire i,j;
    not_ not1(s,i);
    not_ctrl_ nif1(a,i,j);
    not_ctrl_ nif2(b,s,j);
    not_ctrl_ nif3(j,t,w);
endmodule
```

TESTBENCH:

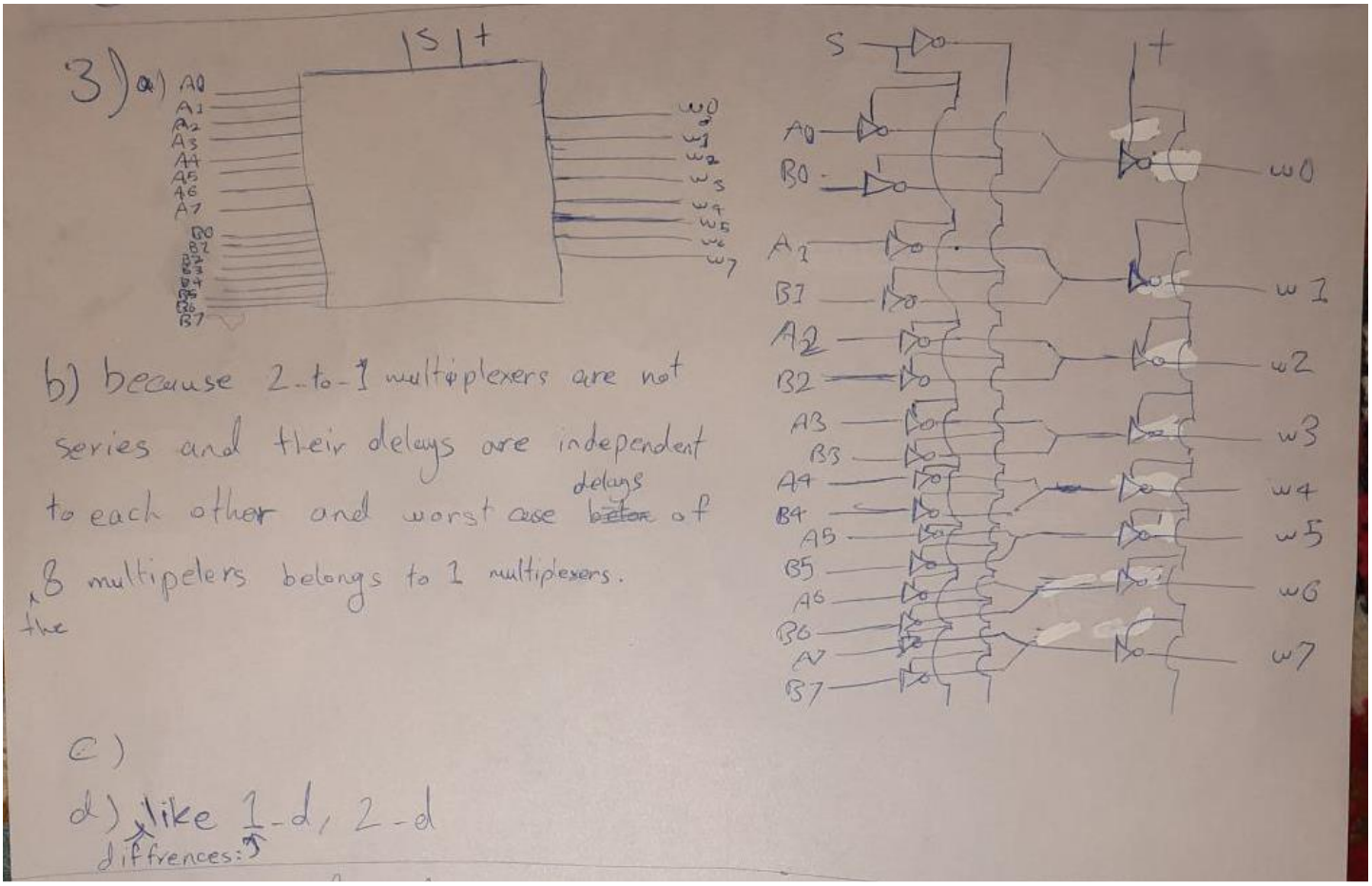
```
`timescale 1ns/1ns
module CA2_E1_TB ();
    logic aa=0,bb=1,ss=0,tt=1;
    wire ww;
    CA2_E1 eee(aa,bb,ss,tt,ww);
    initial begin
    #1000 ss=1;
    #1000 ss=0;
    #500 bb=0;
    #500 tt=0;
    #1000 $stop;
    end
endmodule
```



```
`timescale 1ns/1ns
module not_ (input c,output w2);
    supply1 Vdd;
    supply0 Gnd;
    pmos #(4,7,9) T5(w2,Vdd,c);
    nmos #(3,5,7) T6(w2,Gnd,c);
endmodule
module not_ctrl_ (input c,EN1,output w1);
    wire i1,i2,i3;
    supply1 Vdd;
    supply0 Gnd;
    pmos #(4,7,9) T1(i2,Vdd,c);
    pmos #(4,7,9) T2(w1,i2,i1);
    pmos #(4,7,9) T3(i1,Vdd,EN1);
    nmos #(3,5,7) T4(i1,Gnd,EN1);
    nmos #(3,5,7) T5(w1,i3,EN1);
    nmos #(3,5,7) T6(i3,Gnd,c);
endmodule
module CA2_E1 (input a,b,s,t,output w);
    wire i,j;
    not_ not1(s,i);
    not_ctrl_ nif1(a,i,j);
    not_ctrl_ nif2(b,s,j);
    not_ctrl_ nif3(j,t,w);
endmodule
module CA2_E2 (input a,b,c,d,s,t,output w);
    wire i;
    not_ not1(t,i);
    CA2_E1 multi1(a,b,s,i,w);
    CA2_E1 multi2(c,d,s,t,w);
endmodule
```

TESTBENCH:

```
`timescale 1ns/1ns
module CA2_E2_TB ();
    logic aa=0,bb=1,cc=1,dd=1,ss=0,tt=0;
    wire ww;
    CA2_E2 eee(aa,bb,cc,dd,ss,tt,ww);
    initial begin
    #1000 ss=1;
    #1000 ss=0;
    #1000 $stop;
    end
endmodule
```



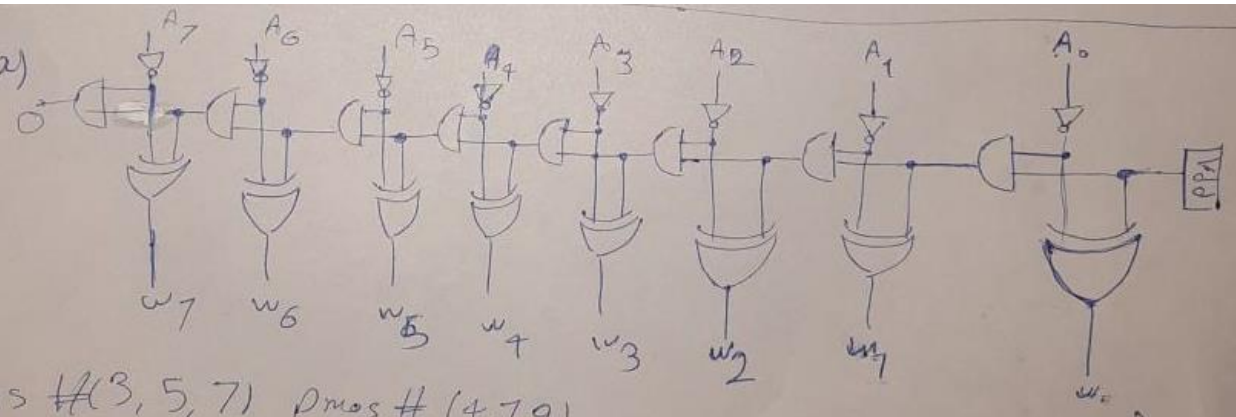

```
`timescale 1ns/1ns
module not_ (input c,output w2);
    supply1 Vdd;
    supply0 Gnd;
    pmos #(4,7,9) T5(w2,Vdd,c);
    nmos #(3,5,7) T6(w2,Gnd,c);
endmodule
module not_ctrl_ (input c,EN1,output w1);
    wire i1,i2,i3;
    supply1 Vdd;
    supply0 Gnd;
    pmos #(4,7,9) T1(i2,Vdd,c);
    pmos #(4,7,9) T2(w1,i2,i1);
    pmos #(4,7,9) T3(i1,Vdd,EN1);
    nmos #(3,5,7) T4(i1,Gnd,EN1);
    nmos #(3,5,7) T5(w1,i3,EN1);
    nmos #(3,5,7) T6(i3,Gnd,c);
endmodule
module CA2_E1 (input a,b,s,t,output w);
    wire i,j;
    not_ not1(s,i);
    not_ctrl_ nif1(a,i,j);
    not_ctrl_ nif2(b,s,j);
    not_ctrl_ nif3(j,t,w);
endmodule
```

```
module CA2_E3 (input [0:7]a,[0:7]b,input s,t,output [0:7]w);
    CA2_E1 e0(a[0],b[0],s,t,w[0]);
    CA2_E1 e1(a[1],b[1],s,t,w[1]);
    CA2_E1 e2(a[2],b[2],s,t,w[2]);
    CA2_E1 e3(a[3],b[3],s,t,w[3]);
    CA2_E1 e4(a[4],b[4],s,t,w[4]);
    CA2_E1 e5(a[5],b[5],s,t,w[5]);
    CA2_E1 e6(a[6],b[6],s,t,w[6]);
    CA2_E1 e7(a[7],b[7],s,t,w[7]);
endmodule
module CA2_E3_2 (input [0:7]a,[0:7]b,input s,t,output [0:7]w);
    assign w= ~s ? a : b ;
endmodule
```

testbench:

```
`timescale 1ns/1ns
module CA2_E3_TB ();
    logic [0:7] aa,bb;
    logic tt=1,ss=0;
    assign aa=8'b01110001;
    assign bb=8'b10100111;
    wire [0:7] ww;
    wire [7:0] wwasign;
    CA2_E3 eee(aa,bb,ss,tt,ww);
    CA2_E3_2 eeee(aa,bb,ss,tt,wwasign);
    initial begin
    #1000 ss=1;
    #1000 ss=0;
    #1000 $stop;
    end
endmodule
```

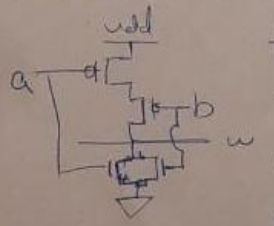
4) a)



nmos #(3, 5, 7) pmos #(4,7,9)

b)

nor → $w = \bar{a} \cdot \bar{b}$

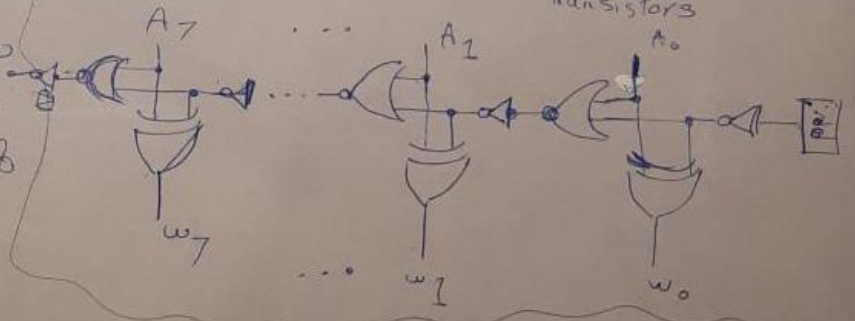


to 1: max(2×4, 7) = 8

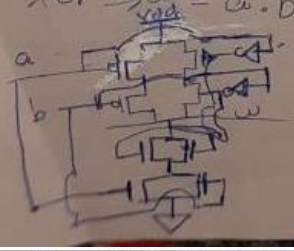
to 0: max(2×9, 5) = 18

nor #(8, 18)

we want to draw with lowest transistors



xor → $w = \bar{a} \cdot \bar{b} + ab$

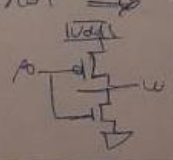


to 1: max(2×4 + 7, 2×7) = 15

to 0: max(~~9~~ + 2×9, 2×5) = ~~25~~ 25

not → to 1: 7

to 0: 9

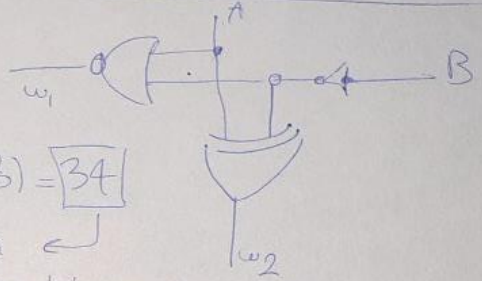


not #(7, 9)

notif1 #(11, 18, 6)

8801994 61

worst case delay for a slice :



AB : 00 → 01 delay = max (9+25, 9+8) = 34

in one slice we can chang A

so delay = max(18, 25) = 25ns

c)

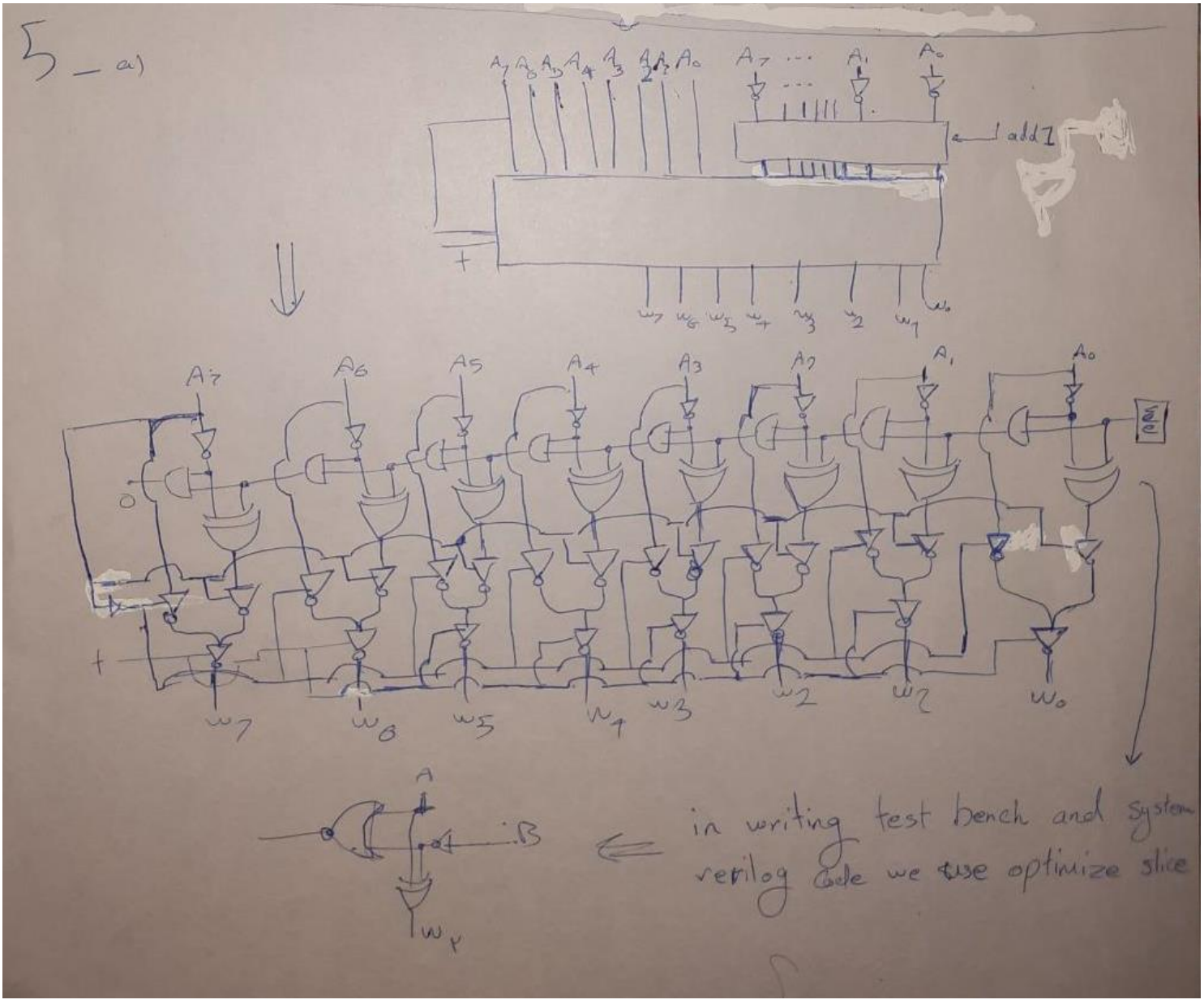
verify : the testbench shows that max delay of 1 slice is 34ns

d) the max delay ~~we~~ hard calculate by hand is 8×34 but in testbench is 200ns

```
`timescale 1ns/1ns
module slice (input a,b,output w2,w1);
    wire i;
    not #(7,9) not1(i,b);
    xor #(15,25) xor1(w2,a,i);
    nor #(8,18) nor1(w1,a,i);
endmodule
module complimentor8_bit (input [7:0] a,output [7:0] w);
    supply1 Vdd;
    wire i[7:0];
    slice slice0(a[0],Vdd,w[0],i[0]);
    slice slice1(a[1],i[0],w[1],i[1]);
    slice slice2(a[2],i[1],w[2],i[2]);
    slice slice3(a[3],i[2],w[3],i[3]);
    slice slice4(a[4],i[3],w[4],i[4]);
    slice slice5(a[5],i[4],w[5],i[5]);
    slice slice6(a[6],i[5],w[6],i[6]);
    slice slice7(a[7],i[6],w[7],i[7]);
endmodule
```

testbench:

```
`timescale 1ns/1ns
module CA2_E4_TB ();
    logic [0:7] aa;
    assign aa=8'b10000010;
    wire [7:0] ww1,ww2;
    complimentor8_bit eee(aa,ww1);
    assign #(34*8) ww2 = ~aa + 1'b1;
    initial begin
    #1000 $stop;
    end
endmodule
module CA2_E4_TB_1slice ();
    logic aa=0,bb=0;
    wire ww1,ww2;
    slice eee(aa,bb,ww2,ww1);
    initial begin
    #1000 bb=1;
    #500 bb=0;
    #500 aa=1;
    #1000 $stop;
    end
endmodule
```



b) 2_to-1 multiplexer #(39, 43, 16)

worst delay of two's complentor section is 200

and it occured when 10000001 entend so we enter 10000001

→ 200 + 14 + 18 = 232ns

تفاوت تعداد گیت یوسیس و برای ما:

برای ما:3*8+3*8=48

یوسیس:12+25+12+1=50

تفاوت 2 تا

```
`timescale 1ns/1ns
module not_ (input c,output w2);
    supply1 Vdd;
    supply0 Gnd;
    pmos #(4,7,9) T5(w2,Vdd,c);
    nmos #(3,5,7) T6(w2,Gnd,c);
endmodule
module not_ctrl_ (input c,EN1,output w1);
    wire i1,i2,i3;
    supply1 Vdd;
    supply0 Gnd;
    pmos #(4,7,9) T1(i2,Vdd,c);
    pmos #(4,7,9) T2(w1,i2,i1);
    pmos #(4,7,9) T3(i1,Vdd,EN1);
    nmos #(3,5,7) T4(i1,Gnd,EN1);
    nmos #(3,5,7) T5(w1,i3,EN1);
    nmos #(3,5,7) T6(i3,Gnd,c);
endmodule
module CA2_E1 (input a,b,s,t,output w);
    wire i,j;
    not_ not1(s,i);
    not_ctrl_ nif1(a,i,j);
    not_ctrl_ nif2(b,s,j);
    not_ctrl_ nif3(j,t,w);
endmodule
module CA2_E3 (input [7:0]a,[7:0]b,s,t,output [7:0]w);
    CA2_E1 e0(a[0],b[0],s,t,w[0]);
    CA2_E1 e1(a[1],b[1],s,t,w[1]);
    CA2_E1 e2(a[2],b[2],s,t,w[2]);
    CA2_E1 e3(a[3],b[3],s,t,w[3]);
    CA2_E1 e4(a[4],b[4],s,t,w[4]);
    CA2_E1 e5(a[5],b[5],s,t,w[5]);
    CA2_E1 e6(a[6],b[6],s,t,w[6]);
    CA2_E1 e7(a[7],b[7],s,t,w[7]);
endmodule
module slice (input a,b,output w2,w1);
    wire i;
    not #(7,9) not1(i,b);
    xor #(15,25) xor1(w2,a,i);
    nor #(8,18) nor1(w1,a,i);
endmodule
```

```
module complimentor8_bit (input [7:0] a,output [7:0] w);
    supply1 Vdd;
    wire i[7:0];
    slice slice0(a[0],Vdd,w[0],i[0]);
    slice slice1(a[1],i[0],w[1],i[1]);
    slice slice2(a[2],i[1],w[2],i[2]);
    slice slice3(a[3],i[2],w[3],i[3]);
    slice slice4(a[4],i[3],w[4],i[4]);
    slice slice5(a[5],i[4],w[5],i[5]);
    slice slice6(a[6],i[5],w[6],i[6]);
    slice slice7(a[7],i[6],w[7],i[7]);
endmodule
module abs (input [7:0] aa,output [7:0] ww);
    wire [7:0] ii;
    supply1 Vdd;
    complimentor8_bit ee(aa,ii);
    CA2_E3 eee(aa,ii,aa[7],Vdd,ww);
endmodule
module abs_assign (input [7:0] aa,output [7:0] ww);
    assign #(8*25+32) ww = aa[7] ? ~aa + 1'b1 : aa;
endmodule
```

testbench:

```
`timescale 1ns/1ns
module CA2_E5_TB ();
    logic [7:0] aa;
    assign aa=8'b10000001;
    wire [7:0] ww1,ww2;
    abs eee(aa,ww1);
    abs_assign eeee(aa,ww2);
    initial begin
    #1000 $stop;
    end
endmodule
```

Yosys:

```
/* Generated by Yosys 0.7 (git sha1 61f6811, i686-w64-mingw32.static-gcc 4.9.3 -Os) */

module abs_assign(aa, ww);
  wire _000_;
  wire _001_;
  wire _002_;
  wire _003_;
```

```
wire _004_;
wire _005_;
wire _006_;
wire _007_;
wire _008_;
wire _009_;
wire _010_;
wire _011_;
wire _012_;
wire _013_;
wire _014_;
wire _015_;
wire _016_;
wire _017_;
wire _018_;
wire _019_;
wire _020_;
wire _021_;
wire _022_;
wire _023_;
wire _024_;
wire _025_;
wire _026_;
wire _027_;
wire _028_;
wire _029_;
wire _030_;
wire _031_;
wire _032_;
wire _033_;
wire _034_;
wire _035_;
wire _036_;
wire _037_;
wire _038_;
wire _039_;
wire _040_;
wire _041_;
wire _042_;
wire _043_;
wire _044_;
wire _045_;
wire _046_;
wire _047_;
wire _048_;
wire _049_;
wire _050_;
wire _051_;
wire _052_;
wire _053_;
```

```
wire _054_;
wire _055_;
wire _056_;
wire _057_;
wire _058_;
wire _059_;
wire _060_;
wire _061_;
wire _062_;
wire _063_;
wire _064_;
wire _065_;
wire _066_;
wire _067_;
wire _068_;
wire _069_;
wire _070_;
wire _071_;
wire _072_;
wire _073_;
wire _074_;
wire _075_;
input [7:0] aa;
output [7:0] ww;
NOT _076_ (
  .A(_018_),
  .Y(_023_)
);
NAND _077_ (
  .A(_070_),
  .B(_023_),
  .Y(_024_)
);
NAND _078_ (
  .A(_070_),
  .B(_027_),
  .Y(_026_)
);
NOR _079_ (
  .A(_070_),
  .B(_027_),
  .Y(_028_)
);
NOR _080_ (
  .A(_028_),
  .B(_023_),
  .Y(_030_)
);
NAND _081_ (
  .A(_030_),
```

```
  .B(_026_),
  .Y(_031_)
);
NAND _082_ (
  .A(_031_),
  .B(_024_),
  .Y(_075_)
);
NOR _083_ (
  .A(_030_),
  .B(_019_),
  .Y(_033_)
);
NOT _084_ (
  .A(_019_),
  .Y(_035_)
);
NOT _085_ (
  .A(_030_),
  .Y(_036_)
);
NOR _086_ (
  .A(_036_),
  .B(_035_),
  .Y(_037_)
);
NOR _087_ (
  .A(_037_),
  .B(_033_),
  .Y(_020_)
);
NOT _088_ (
  .A(_021_),
  .Y(_039_)
);
NOT _089_ (
  .A(_027_),
  .Y(_041_)
);
NOT _090_ (
  .A(_070_),
  .Y(_042_)
);
NAND _091_ (
  .A(_042_),
  .B(_041_),
  .Y(_043_)
);
NOR _092_ (
  .A(_043_),
```

```
  .B(_019_),
  .Y(_045_)
);
NOR _093_ (
  .A(_045_),
  .B(_023_),
  .Y(_046_)
);
NOT _094_ (
  .A(_046_),
  .Y(_047_)
);
NOR _095_ (
  .A(_047_),
  .B(_039_),
  .Y(_048_)
);
NOR _096_ (
  .A(_046_),
  .B(_021_),
  .Y(_049_)
);
NOR _097_ (
  .A(_049_),
  .B(_048_),
  .Y(_022_)
);
NOT _098_ (
  .A(_025_),
  .Y(_050_)
);
NAND _099_ (
  .A(_028_),
  .B(_035_),
  .Y(_051_)
);
NOR _100_ (
  .A(_051_),
  .B(_021_),
  .Y(_052_)
);
NOR _101_ (
  .A(_052_),
  .B(_023_),
  .Y(_053_)
);
NOT _102_ (
  .A(_053_),
  .Y(_054_)
);
```

```
NOR _103_ (
  .A(_054_),
  .B(_050_),
  .Y(_055_)
);
NOR _104_ (
  .A(_053_),
  .B(_025_),
  .Y(_056_)
);
NOR _105_ (
  .A(_056_),
  .B(_055_),
  .Y(_029_)
);
NOT _106_ (
  .A(_032_),
  .Y(_057_)
);
NAND _107_ (
  .A(_052_),
  .B(_050_),
  .Y(_058_)
);
NAND _108_ (
  .A(_058_),
  .B(_018_),
  .Y(_059_)
);
NOR _109_ (
  .A(_059_),
  .B(_057_),
  .Y(_061_)
);
NOT _110_ (
  .A(_059_),
  .Y(_062_)
);
NOR _111_ (
  .A(_062_),
  .B(_032_),
  .Y(_063_)
);
NOR _112_ (
  .A(_063_),
  .B(_061_),
  .Y(_034_)
);
NOT _113_ (
  .A(_038_),
```

```
  .Y(_064_)
);
NAND _114_ (
  .A(_045_),
  .B(_039_),
  .Y(_065_)
);
NOR _115_ (
  .A(_065_),
  .B(_025_),
  .Y(_066_)
);
NAND _116_ (
  .A(_066_),
  .B(_057_),
  .Y(_067_)
);
NAND _117_ (
  .A(_067_),
  .B(_018_),
  .Y(_068_)
);
NOR _118_ (
  .A(_068_),
  .B(_064_),
  .Y(_069_)
);
NOR _119_ (
  .A(_058_),
  .B(_032_),
  .Y(_071_)
);
NOR _120_ (
  .A(_071_),
  .B(_023_),
  .Y(_072_)
);
NOR _121_ (
  .A(_072_),
  .B(_038_),
  .Y(_073_)
);
NOR _122_ (
  .A(_073_),
  .B(_069_),
  .Y(_040_)
);
NAND _123_ (
  .A(_064_),
  .B(_018_),
```

```
    .Y(_074_)
  );
  NOR _124_ (
    .A(_074_),
    .B(_067_),
    .Y(_044_)
  );
  BUF _125_ (
    .A(_027_),
    .Y(_060_)
  );
  assign _018_ = aa[7];
  assign _027_ = aa[0];
  assign ww[0] = _060_;
  assign _070_ = aa[1];
  assign ww[1] = _075_;
  assign _019_ = aa[2];
  assign ww[2] = _020_;
  assign _021_ = aa[3];
  assign ww[3] = _022_;
  assign _025_ = aa[4];
  assign ww[4] = _029_;
  assign _032_ = aa[5];
  assign ww[5] = _034_;
  assign _038_ = aa[6];
  assign ww[6] = _040_;
  assign ww[7] = _044_;
endmodule
```