178 Exam Review

4 Questions - 15 mins each

Topics

1. Mental Synthesis - (Here's a circuit, show circuit or vice-versa)
2. Mental Simulation -
3. Static Timing Analysis - Fwd.) In class
4. Static Timing Analysis - Rev.
5. Debugging - Verilog syntax errors or logical errors
6. Constraints Fill-in-the-Blank
7. Verilog Fill-in-the-Blank
8. Counter Based Design (Lab 2/4 type of problem)

ON EXAM

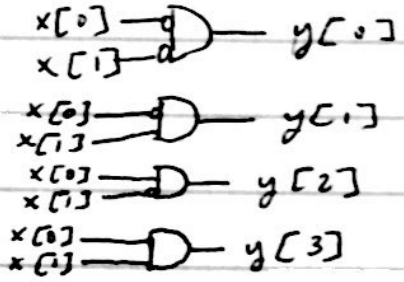
[redacted]

Mental Synthesis

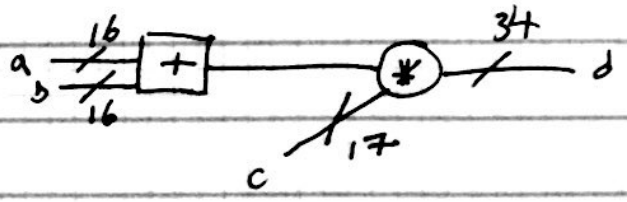
Ex 1

```
wire [1:0] x;
reg [3:0] y;
always @*
begin
    y = 4'b0001 << x;
end
```

Sol 1



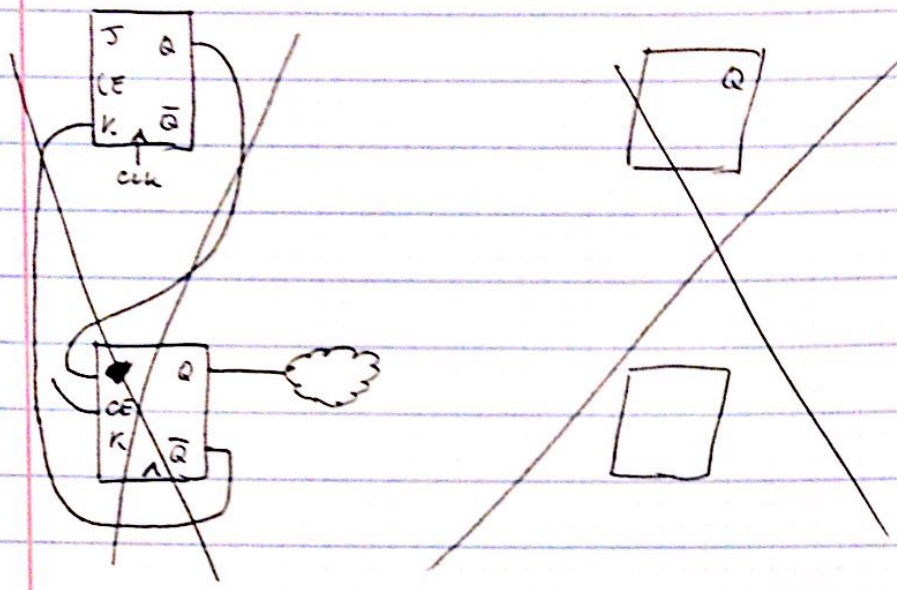
Ex 2



Sol 2:

```
wire [15:0] a, b;
wire [16:0] c;
wire [33:0] d;
assign d = (a+b) * c;
```

Static Timing Analysis

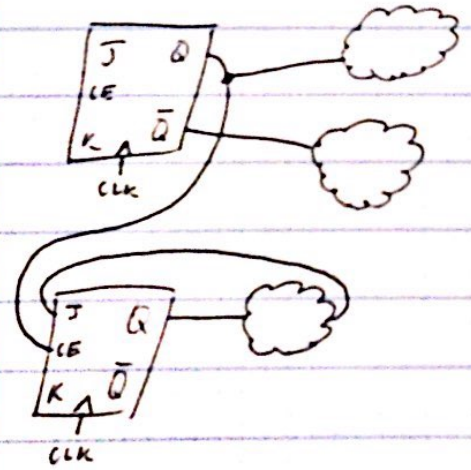


— Two FFs with random cloud types of logic

Q1: How many paths?

Q2: Will it work?

Q3: Assuming it works, ~~work~~ what is T_{min}?



Q2:

Add up all delays through the path and check the hold requirement of destination

Path delays $\geq T_H$

Q3. Identify longest path intuitively (any path with ~~7~~ ns delay here)

setup time

$2 + 7 + 2 + 1 = 12\,ns = T_{min}$

$2 + 7 + 1 = 10\,ns$

Debugging

- This question is a bit oversimplified

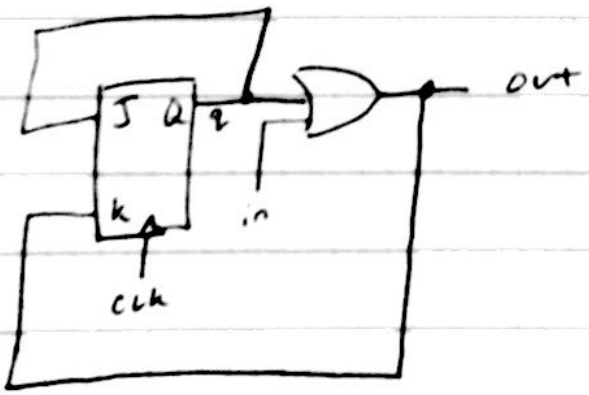
iToilet: Supposed to flush when nobody is sitting and it is full

code (given):

```
wire full;
wire nopersoni
wire flushi
always @*
begin
    assign flush = full ^ nopersoni
end
```

procedural block { can't use wires in procedural block

Verilog FITB (Fill-in-the-Blanks)

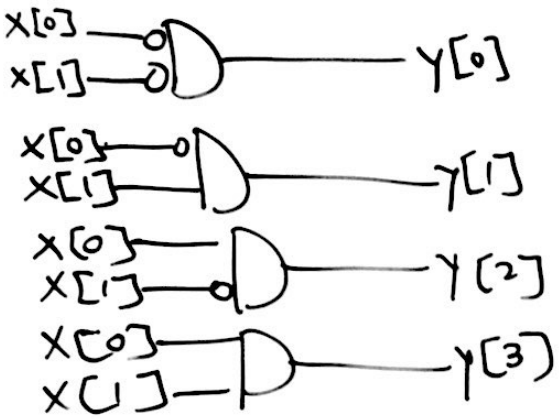


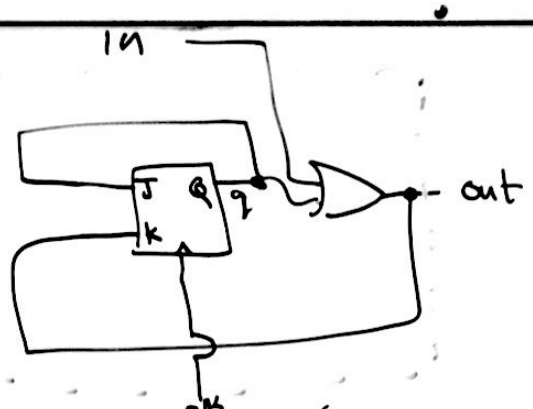

```
module skeleton(
___ ___ in,
output ___ out,
___ ___ clk);

always @ _____
case({q, ___})
0: q
1: q
2: q
3: q

assign out = _____;
```

FF eqn
JK FF eqn
SR FF eqn
D

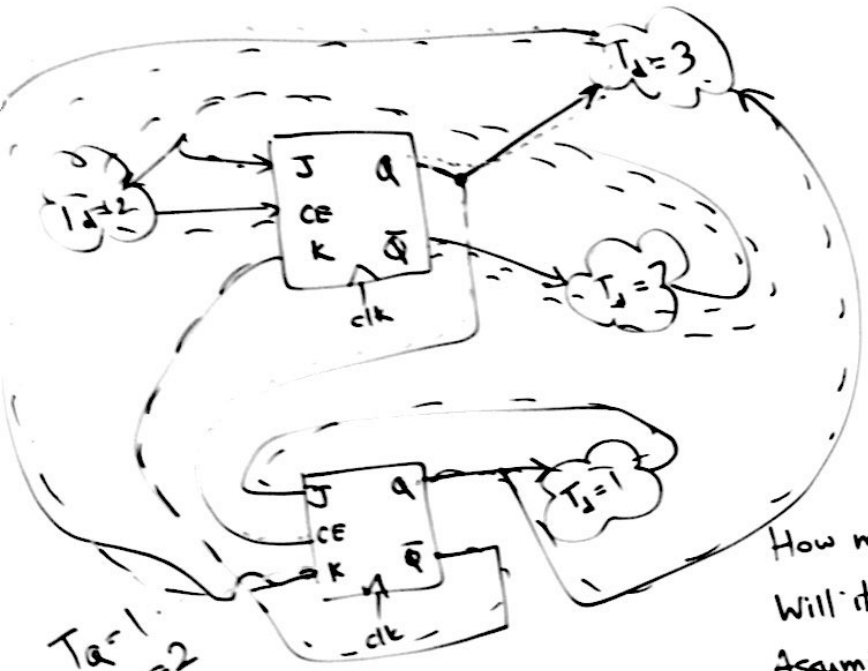
; Toilet
it supposed to flush when full and no person is sitting

```
wire full;
wire noperson;
wire flush;

always @*
  begin
    assign flush <= full ^ noperson;
  end
```






```
module whatever (
__ __ in,
output __ out,
__ __ clk );

wire [15:0] a, b;
wire [16:0] c;
wire [33:0] d;

assign d = c * (a + b);

always @ _____
begin
case ( {q, __} )
  0: q __;
  1: q __;
  2: q __;
  3: q __;
endcase
end

assign out = _____;

endmodule
```

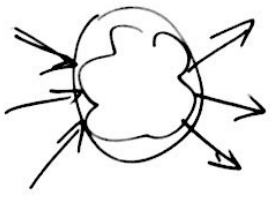


$T_{CQ} = 1$
$T_{CQD} = 2$
$T_{SUJ} = 1$
$T_{HJ} = 1$
$T_{SUCE} = 0$
$T_{HCE} = 0$
$T_{SUR} = 2$
$T_{HR} = 2$
$T_W = 0$

(A)

$1 + 3 \geq 2$ yes
$1 \geq 0$ yes
$2 + 7 \geq 1$ yes
$2 + 7 + 2 \geq 0$ yes

How many paths? 7
Will it work? yes
Assuming it works, what is Tmin 12 ns



D = copy (data)
T = toggle
SR = set / reset

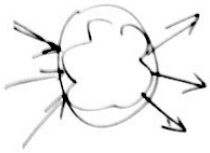
2 yes
yes
1 yes
2 ≥ 0 yes

1 + 1 ≥ 1 yes
1 + 3 ≥ 2 yes
2 ≥ 2 yes

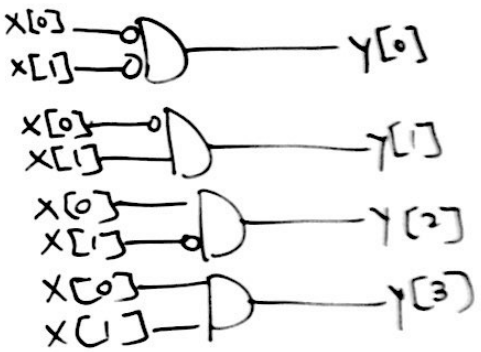
J K
set
clear

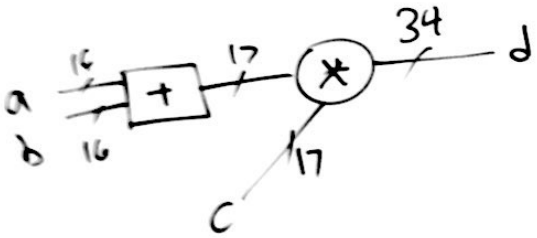
Tmin 12 ns

Setup Check
2 + 7 + 2 + 1 = 12 ns slower
2 + 7 + 1 = 10 ns faster



```
wire [1:0] x;
reg [3:0] y;
always @*
begin
    y = 4'b0001 << x;
end
```






```
wire [15:0] a, b;
wire [16:0] c;
wire [33:0] d;

assign d = c * (a + b
```

1. Mental Synthesis ~ 5 simple xlates
2. Mental Simulation ~ few simple xlates
3. Static Timing Analysis – fwd.
4. Static Timing Analysis – rev.
5. Debugging
6. Constraints FITB
7. Verilog FITB
8. Counter Based Design

x	y
00	0001
01	0010
10	0100
11	1000

1. Mental Synthesis ~ 5 simple xlates
2. Mental Simulation ~ few simple xlates
3. Static Timing Analysis – fwd.
4. Static Timing Analysis – rev.
5. Debugging
6. Constraints FITB
7. Verilog FITB
8. Counter Based Design

x
00
01
10
11