SOLUTION *Your Name (first last)* *← Name of person on left (or aisle)*	**UC Berkeley EECS151** **EECS251A** **Fall 2021 Final Exam** Room _____ Lab *TA name*	_____ *SID* *Name of person on right (or aisle) →*

Fill in the correct circles & squares completely…like this: ● (select ONE), and ⬛(select ALL that apply)

Fill in your student ID at the top of every page.

Question	1	2	3	4	5	6	7	8	Total
Clobber	MT1	MT1	MT2	MT2	-	MT2	-	-	
Minutes	15	35	10	25	30	20	15	15	**165**
Max Points (151)	15	34	8	14	18	14	18	13	**134**
Max Points (251A)	15	34	8	18	24	14	18	13	**144**
Points									

SID : _____

1) *FSM* (15 points, 15 minutes)

a) (5 pts) We would like to design a finite-state machine (FSM) that eliminates single 1s and 0s in a bit pattern. For example if the input pattern is
00100110111, it should be replaced with an output pattern:
00000011111.
An example FSM for this function is shown in Figure 1. It assumes that the pattern always starts with a leading 0. The output is delayed by one cycle from the corresponding input, because it takes a cycle to determine that it is not a single 1 or 0.
In the state transition diagram below, all inputs are labeled, but some outputs are unlabeled. Please complete the diagram.

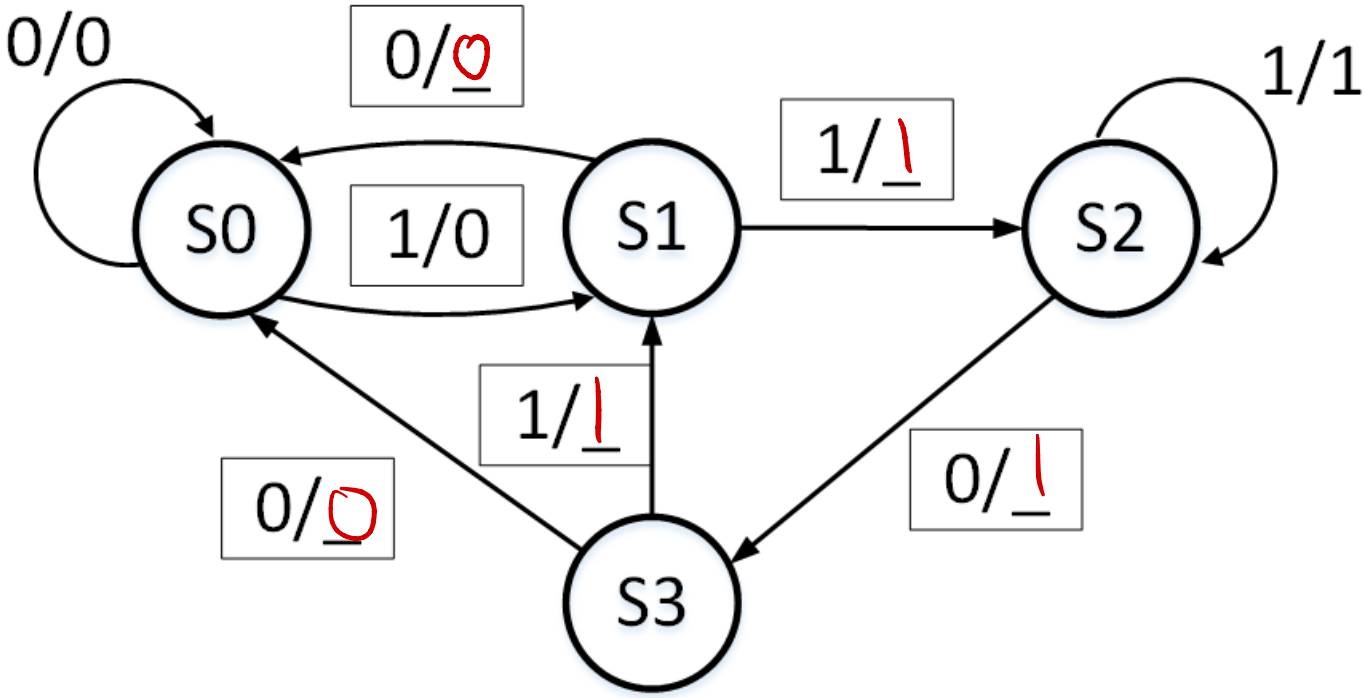


Figure 1

SID : _____

b) (10 pts) Assume that the states are binary, little-endian encoded S0 = 00, S1 = 01, S2 =10, S3 = 11. Derive the minimal logic that computes the new state {s1, s0} from the previous state {ps1, ps0} and the single-bit input, in.

S1

in \ $P_{S_1}P_{S_0}$	00	01	11	10
0	0	0	0	1
1	0	1	0	1

S0

in \ $P_{S_1}P_{S_0}$	00	01	11	10
0	0	0	0	1
1	1	0	1	0

s1(ps1, ps0, in) = $P_{S_1}\overline{P_{S_0}} + \overline{P_{S_1}}\,P_{S_0}\,in$

s0(ps1, ps0, in) = $P_{S_1}\overline{P_{S_0}}\,\overline{in} + P_{S_1}P_{S_0}\,in + \overline{P_{S_1}}\,\overline{P_{S_2}}\,in$

2) *Verilog* (34 points, 35 minutes)

a) (4 pts) Are the circuits inferred by the two blocks always equivalent? Mark Yes or No. Incorrect answers carry negative credit.

i)

```
wire [9:0] a, b;
wire [10:0] c;
assign c = a + b;
```

```
wire [9:0] a, b;
wire [10:0] c;
assign c = $signed(a) + $signed(b)
```

● Yes ○ No 'signed' and 'unsigned' addition infer the same circuit

ii)

```
wire [7:0] a, c;
wire [2:0] b;
assign c = a >>> b;
```

```
wire [7:0] a, c;
wire [2:0] b;
assign c = $signed(a) >>> b;
```

○ Yes ● No Verilog infers arithmetic right shift *only* if the first operand is signed

iii)

```
wire a;
assign a = 1'bx;
```

```
wire a;
assign a = 1'b0;
```

○ Yes ● No Assigning a wire to 'x' allows the synthesis tool to arbitrarily set the value of the wire

iv)

```
wire [1:0] a;
reg [3:0] b;
always @(*) begin
  case(a)
    2'b1x: b = 4;
    2'b01: b = 3;
    2'b0x: b = 2;
  endcase
end
```

```
wire [1:0] a;
reg [3:0] b;
always @(*) begin
  b = 3;
  if (a == 2'b01)
    b = 2;
  else if (a == 2'b00)
    b = 2;
  else if (a == 2'b10)
    b = 4;
end
```

Assume that a case item with an "x" is matched when the case expression is 0 *or* 1. e.g. 3'b11x matches 3'b110 and 3'b111.

○ Yes ● No There is a mismatch between the 2 blocks when a = 2'b11 (first block sets b = 4 and second block sets b = 3).

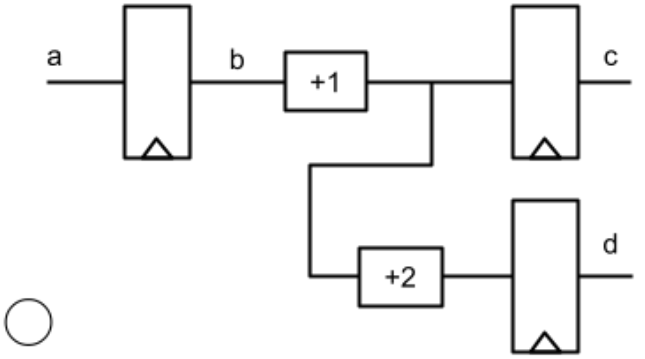
SID : _____

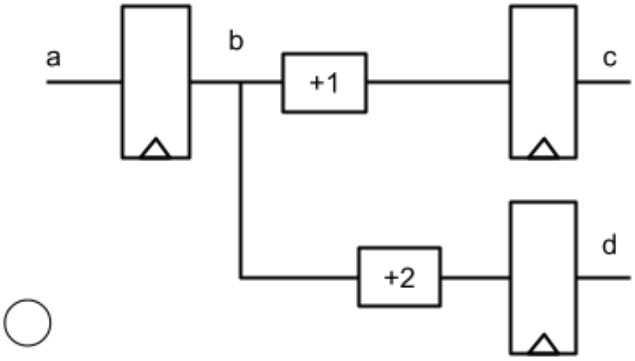
b) (2 pts) What circuit does this Verilog synthesize to?

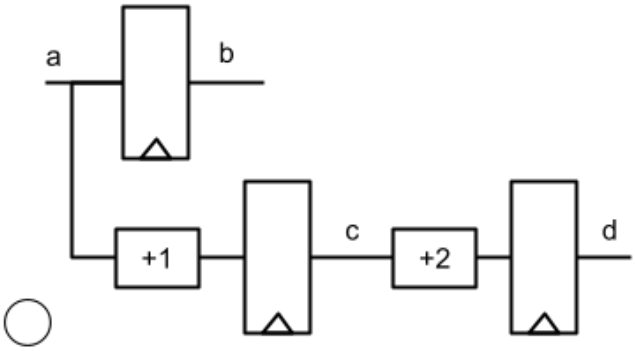
```
wire [1:0] a;
wire clk;
reg [1:0] b, c, d;

always @(posedge clk) begin
    b <= a;
    c = b + 2'b1;
    d <= c + 2'b2;
end
```

Answer: The first circuit. c is assigned using blocking assignment, so the 'new' value of c is used to drive the logic that assigns d instead of the output of the c register.







SID : _____

c) (8 pts) We want to find the minimum unsigned entry in an async read RAM, by iterating through one entry per clock cycle. Fill in the values of the blank spaces to implement the circuit.

```
reg [7:0] ram [0:15]; // Assume the RAM is preloaded with data
reg [(a):0] addr;
reg [(b):0] minimum;

always @(posedge clk) begin
    if (rst)
        addr <= (c);
    else
        addr <= (d);
end

always @(posedge clk) begin
    if (rst)
        minimum <= (e);
    else
        minimum <= (ram[addr] (f) minimum) ? (g) : (h);
end
```

(a)	3 (2^4 = 16 = the number of entries in the RAM) (other answers won't be accepted)
(b)	7 (each RAM entry is 8 bits wide)
(c)	0 or 1 or 15 (or anything between 0-15)
(d)	addr + 1 or addr - 1
(e)	ram[0] or 8'hff or ram[any index]
(f)	<
(g)	ram[addr]
(h)	minimum

(f) can also be >, in which case (g) and (h) will be flipped.

SID : _____

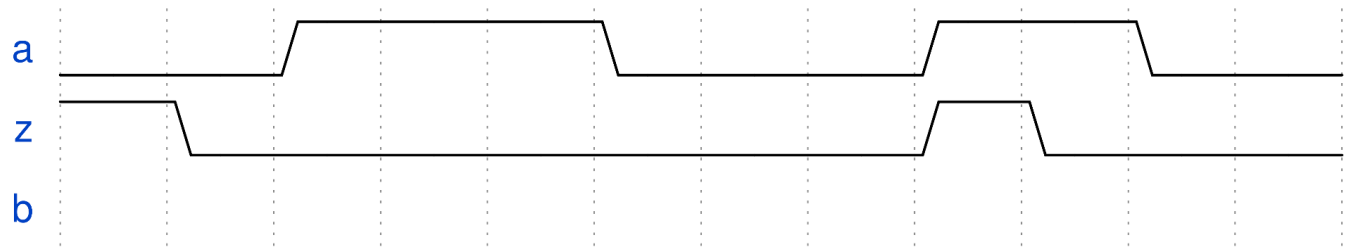
d) (4 pts) Say you have a 1 cycle ALU with 2 N-bit inputs and M distinct operations. You can simulate the ALU using RTL simulation at a speed of C Hz (i.e. you can simulate C clock cycles in 1 second).

How long does it take to exhaustively test the ALU? Give your answer in terms of N, M, and C. Include units in your answer.

Answer: $(2^N * 2^N * M) / C$ seconds

e) (2 pts) Complete the waveform for the following circuit:

```
wire a, z;
reg b;
always @(*)
    if (z)
        b = a;
```



b should start low from the very first moment (since z is high and thus a propagates to b).
b should stay low until both z and a go high, at which point b goes high.
b should continue to stay high for the rest of the waveform (since when z drops, b holds the value it was last assigned. when a drops later it has no impact on b).

SID : _____

f) (6 pts) Do the following Verilog snippets describe synthesizable synchronous digital logic? Mark yes or no, and if no, give the reason.

```
reg a, b; // assume a and b are the direct outputs of flip-flops
wire clk;
wire k;
reg s;
assign k = (a == s) ? b : a;
always @(posedge clk) begin
    s <= k;
end
```

● Yes O No

If you marked “No”, why? _____

```
reg a, b; // assume a and b are the direct outputs of flip-flops
wire clk;
wire k;
reg s;
assign k = (k == s) ? b : a;
always @(posedge clk) begin
    s <= k;
end
```

O Yes ● No

If you marked “No”, why? Assignment to k forms a combinational loop. Answers that mention self-referential logic or infinite toggling or unstable logic are acceptable.

SID : _____

g) (8 pts) A decoder converts an N-bit integer to a 2^N-bit one-hot encoded signal.

i) Fill in the behavioral description of a decoder:

```
wire [7:0] addr;
wire [255:0] dec;
assign dec = (a) << (b);
```

(a)	256'd1	(b)	addr

ii) Let's design a structural decoder. What is the logic for a 1-bit decoder? Fill in the blanks.

```
wire addr;
wire [1:0] dec;
assign dec[0] = (a);
assign dec[1] = (b);
```

(a)	!addr	(b)	addr

iii) How can you build a 2-bit decoder using 1-bit decoders (dec1bit)? Fill in the blanks.

```
wire addr[1:0];
wire [3:0] dec;
wire [1:0] dec1, dec2;
dec1bit d1 (.addr(addr[0]), .dec(dec1));
dec1bit d2 (.addr(addr[0]), .dec(dec2));
assign dec[1:0] = dec1 & (a);
assign dec[3:2] = dec2 & (b);
```

(a)	{2{!addr[1]}}	(b)	{2{addr[1]}}

SID : _____

h) (6 pts) Does the Verilog snippet exhibit a problem (syntax or elaboration error)? If so, what is the problem?

```
wire a;
assign a = 1'b1;
```

O Problem ● No Problem

If there's a problem, what is it? _____

```
wire a;
always @(*) a = 1'b1;
```

● Problem O No Problem

If there's a problem, what is it? Can't assign a wire inside an always block

```
reg [1:0] a;
wire b;
always @(*)
    a = b ? 2'b1 : 2'b2;
```

● Problem O No Problem

If there's a problem, what is it? The literal 2'b2 is illegal since '2' isn't a binary value

```
wire a, c;
reg b;
always @(*)
    b = a + c;
always @(*)
    b = !a;
```

● Problem O No Problem

If there's a problem, what is it? 'b' is a multi-driven net which is illegal in synthesized Verilog

SID : _____

```
module a(input x, output y);
    always @(*)
        y = !x;
endmodule
```

● Problem O No Problem

If there's a problem, what is it? Output nets are by default wires, which can't be assigned in always blocks. The declaration should be 'output reg y'

```
wire a, z, b;
always @(z)
    a = z ? b : !b;
```

● Problem O No Problem

If there's a problem, what is it? Cannot assign a (wire) inside an always block. Incomplete sensitivity list is also an acceptable problem to mention.

3) *CMOS* (8 points, 10 minutes)

The circuit below can be programmed to compute various logic functions of two variables, A and B, by setting the appropriate values for R1, R2, R3, R4. The supply is VDD and all transistors have a threshold voltage $|V_{Th}|$.

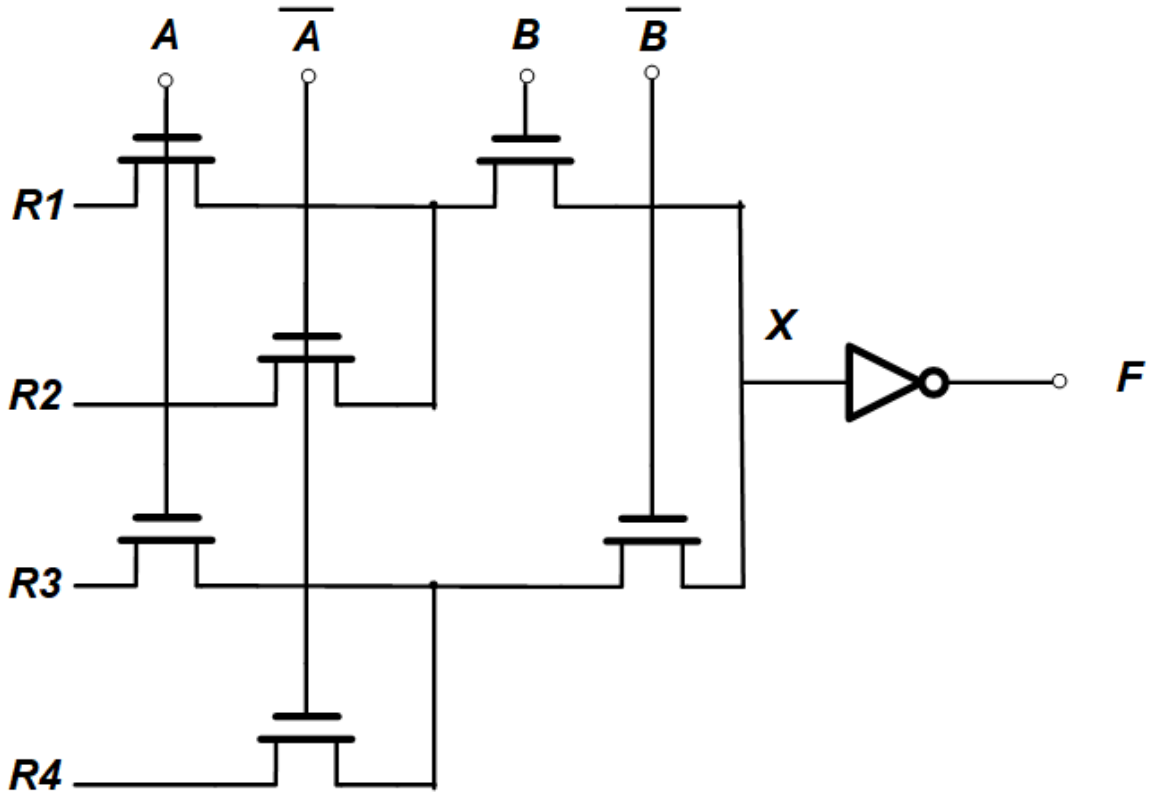


Figure 3.

a) (2 pts) How many different logic functions can this circuit implement?

$2^4 = 16$

Number of functions _____16_____

b) (4 pts) For which logic values of R1, R2, R3, R4 does this circuit implement F = A+B.

$X = \overline{F} = \overline{A + B} = \overline{A} \cdot \overline{B}$

If $\overline{A}$ and $\overline{B}$, then the bottom center and bottom right NMOS will switch on. So, R4 should be 1.

R1	0
R2	0
R3	0
R4	1

c) (2 pts) What is the voltage at node X, if R1, R2, R3, R4, A and B are all at V_{DD}?

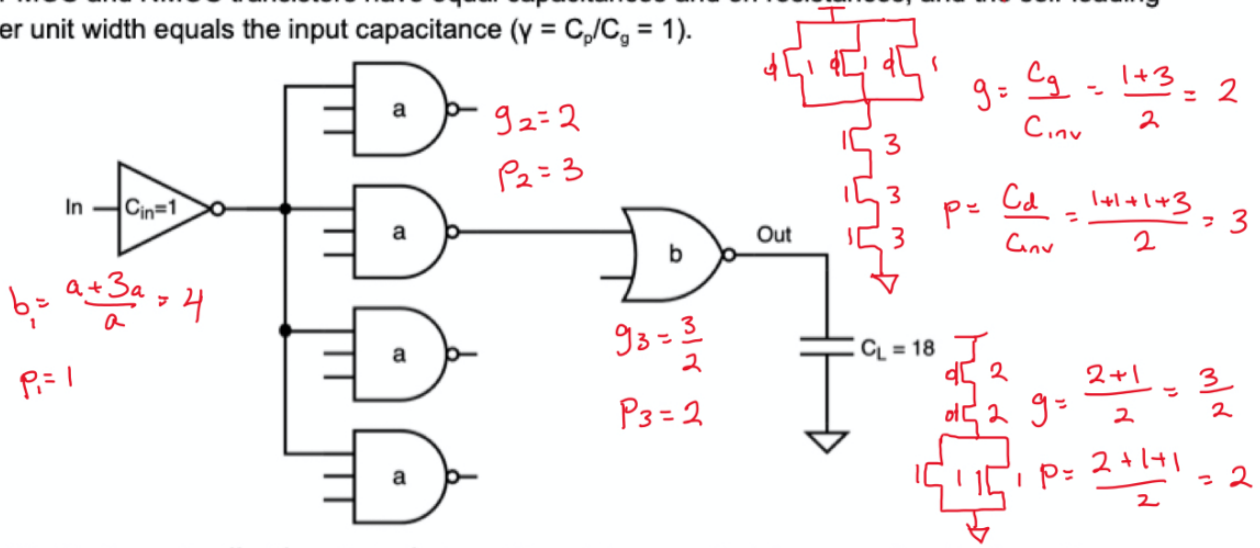
The top 2 NMOS will be switched on. Specifically the one on the right will drive X with $V_{DD} - V_{Th}$.

VX = _____$V_{DD} - V_{Th}$_____

SID : _____

4) *Fast Circuits* (14/18 points, 20/25 minutes)

Consider the following logic path. Cin, a and b are the capacitances of the input pins of their respective gates. Equally sized PMOS and NMOS transistors have equal capacitances and on resistances, and the self-loading capacitance per unit width equals the input capacitance ($\gamma = C_p/C_g = 1$).



a) (4 pts) What is the path effort from In to Out, and the minimum path delay normalized to the unit inverter?

$B = \Pi b_i = 4 \cdot 1 \cdot 1 = 4$

$F = \frac{C_L}{C_{in}} = 18$

$G = \Pi g_i = 1 \cdot 2 \cdot \frac{3}{2} = 3$

$H = BFG = 4 \cdot 18 \cdot 3 = 216$

$h^* = \sqrt[N]{H} = \sqrt[3]{216} = 6$

$P = \Sigma p_i = 1+3+2 = 6$

$D = \frac{t_{p,gate}}{\tau_{inv}} = \Sigma h_i + \Sigma p_i$

$= 3 \cdot 6 + 6 = 24$

Path effort: H = 216 Path delay ($t_{p,gate}/T_{INV}$) = 24

b) (4 pts) Size the gates to minimize the delay from In to Out.

$h^* = b_i f_i g_i \rightarrow$ stage 1: $6 = 4 \cdot \frac{a}{1} \cdot 1 \rightarrow a = \frac{6}{4} = \frac{3}{2}$

$\rightarrow$ stage 3: $6 = 1 \cdot \frac{18}{b} \cdot \frac{3}{2} \rightarrow b = \frac{9}{2}$

check: stage 2: $6 = 1 \cdot \frac{b}{a} \cdot 2 \rightarrow 6 = \frac{9/2}{3/2} \cdot 2$ ✓

Gate sizes: a = 3/2 b = 9/2

SID : _____

c) (4 pts) Suppose you can add as many buffers of any size (buffer = 2 inverters) to this path to minimize the path delay. How many buffers should you add?

ideal # stages approximation = $\log_4(216) = \frac{\ln(216)}{\ln(4)} = 3.88$

test path delays around this # of stages (N)

let x = # of buffers added

H = 216 (unchanged from buffers b/c BFG = 1 for inverters)

$P = P_{1-3} + 2x = 6 + 2x$ (original parasitic delay + 1 per inverter)

$D = N\sqrt[N]{H} + P = N\sqrt[N]{216} + 6 + 2x$

buffers = 0 → N = 3 → D = 24 (same as before)

= 1 → N = 5 → $D = 5\sqrt[5]{216} + 6 + 2 \cdot 1 = 22.65$ ← ** minimum

= 2 → N = 7 → $D = 7\sqrt[7]{216} + 6 + 2 \cdot 2 = 25.09$

Number of buffers = 1

d) (2 pts) To achieve the lowest power design for the minimum delay, the buffers from part c) should be added (mark one):

for low power we want to minimize total gate capacitance

$h_i = b_i f_i g_i = g_i \frac{C_{L,i}}{C_i}$ → highest C_L in design is at output, so select gate with lowest g (inverter) to drive this output

O after the first inverter

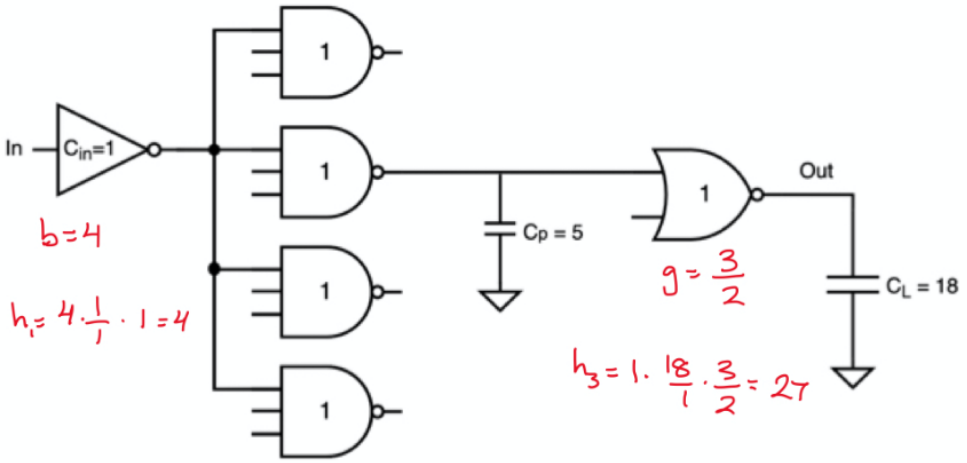
O after the NAND gate

● after the NOR gate

alternatively: recall that logical effort represents how well a gate can drive an output current, so place inverters before largest load (C_L)

SID : _____

e) **(251A only)** (4 pts) You decide to size all the gates the same, with size C_{in}=a=b=1, and fabricate the circuit, only to realize there is a large parasitic capacitance between the second and the third stage of the circuit. The new diagram is shown below. Calculate the delay of the logic path normalized to the unit inverter.



$b = 4$

$h_1 = 4 \cdot \frac{1}{1} \cdot 1 = 4$

$g = \frac{3}{2}$

$h_3 = 1 \cdot \frac{18}{1} \cdot \frac{3}{2} = 27$

$g = 2$

$f = \frac{6}{1}, b = 1$

(or $f = 1, b = \frac{1+5}{1} = 6$)

$h_2 = 1 \cdot \frac{6}{1} \cdot 2 = 12$

$P = 6$ (same as before)

$D = \sum h_i + P$

$= 4 + 12 + 27 + 6$

$= 49$

Path delay ($t_{p,gate}/\tau_{INV}$) = 49

SID : _____

5) *Timing* (18/24 points, 20/30 minutes)

Consider the following circuit diagram. R0, R1, and R2 are rising-edge triggered flip-flops. The maximum and minimum combinational delays are listed for each path. You can assume that t_{clk-q} = 50ps , t_{su} = 75ps , and t_{hold} = 60ps, and the multiplexers have negligible delay. SEL is a single control bit that is stable during any given clock cycle. CLK2 and CLK3 are clock signals derived from CLK1 with some amount of delay, as indicated in Figure 4.

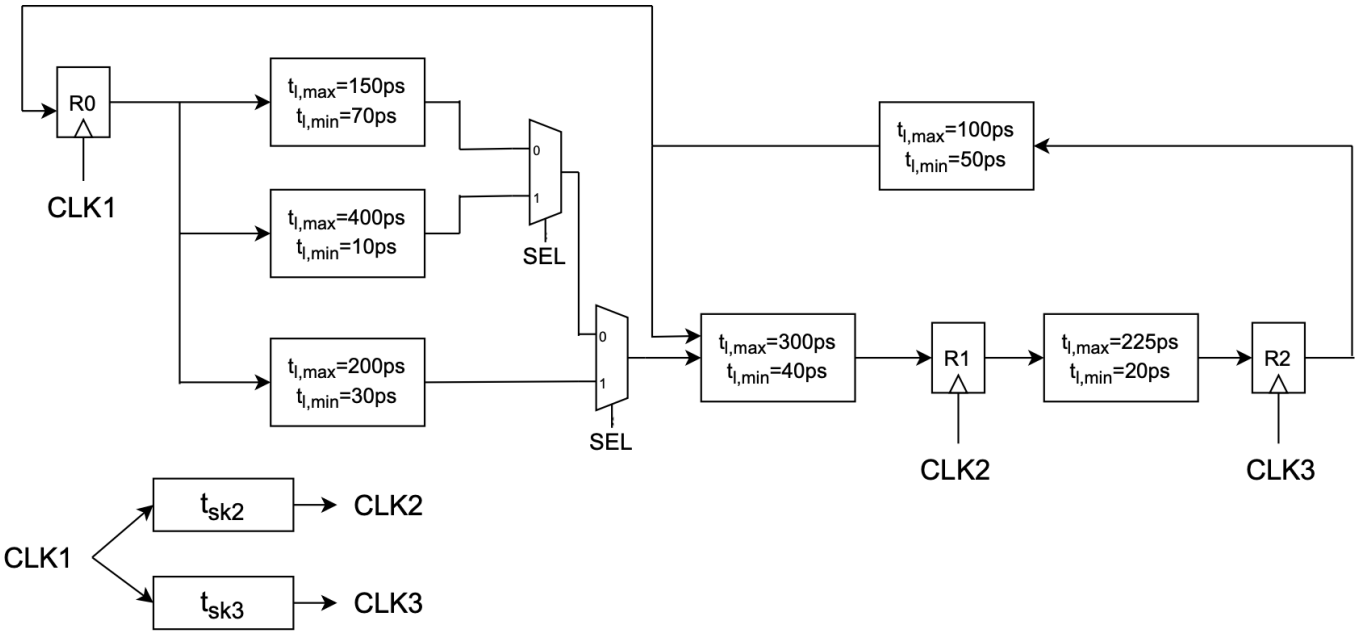


Figure 4.

(a) (6 pts) You may start by assuming that the clock source and clock tree are ideal (tsk2 and tsk3 are both 0). Answer the following questions.

(i) What is the minimum clock period, Tmin, that this circuit may operate at correctly?

Note the architectural false path from the mux select; max path is R0→R1 with SEL=1

$$T_{clk} \geq t_{clk-q} + t_{l,max} + t_{su}$$
$$\geq 50ps + (200ps + 300ps) + 75ps$$
$$\geq 625ps$$

Tmin = 625 ps.

(ii) What is the hold slack for this circuit (a negative hold slack value indicates a violation)?

min path is R2→R3

$$\text{hold slack} = (t_{clk-q} + t_{l,min}) - t_h$$
$$= (50ps + 20ps) - 60ps$$
$$= 10ps$$

Slack = 10 ps.

SID : _____

(b) (6 pts) Now, with a physical clock network, it is found that t_{sk2} = 150ps and t_{sk3} = −100ps . Answer the following questions.

(i) What is the minimum clock period, Tmin , that this circuit may operate at correctly?

min path with skew is R1→R2

$$T_{clk} \geq t_{clk\text{-}q} + t_{l,max} + t_{su} - t_{sk}$$
$$\geq 50ps + 225ps + 75ps - (-100ps - 150ps)$$
$$\geq 600ps$$

Tmin = 600 ps.

(ii) What is the hold slack for this circuit (a negative hold slack value indicates a violation)?

min path with skew is R2→R1

$$hold\ slack = (t_{clk\text{-}q} + t_{l,min}) - (t_h + t_{sk})$$
$$= (50ps + 90ps) - (60ps + (150_{ps} - (-100_{ps})))$$
$$= 140_{ps} - (310_{ps})$$
$$= -170ps$$

Slack = -170 ps.

(c) (6 pts) Now, we will consider the effect of variability in the clock network. The clock source, CLK1, has cycle-to-cycle jitter of 50ps . Furthermore, each of t_{sk2} and t_{sk3} have a variability of ±10% in their delay, (the nominal delay is the same as in part (b)). Answer the following questions.

(i) What is the minimum clock period, Tmin, that this circuit may operate at correctly?

- Same skew as part (b) → take max path and add jitter in worse direction
- Note jitter is on CLK1, so it will affect all clocks the same way
- Also, add skew variability in the worse direction on the max path

skew variability for R1→R2: t_{sk2} = 150 ± 15ps, t_{sk3} = −100 − 10ps

$$T_{min} = T_{min,0} - t_{sk,var} + t_j$$
$$= 600ps + (10ps + 15ps) + 50ps$$
$$= 675ps$$

Tmin = 675 ps.

(ii) What is the hold slack for this circuit (a negative hold slack value indicates a violation)?

- Same hold path as before
- no effect from cycle-to-cycle jitter because all clocks affected the same way
- add skew variation in worse direction

R2→R1

$$hold\ slack = slack_{,0} - t_{sk,var}$$
$$= -170ps - (10ps + 15ps)$$
$$= -195\ ps$$

Tmin = -195 ps.

SID : _____

(d) **(EECS251A Only)** (6 pts) You may now set the values of t_{sk2} and t_{sk3}. What would you set t_{sk2} and t_{sk3} to allow for the shortest clock period possible? What is that Tmin ? You may ignore hold time violations (they can be fixed by adding buffers) and assume that t_{sk2} and t_{sk3} have no variability.

Want to equalize path delays as much as possible, so use skews to redistribute
↳ write constraints and minimize T_{clk}

Constraints: $T_{clk} \geq$

	check
$0 \to 1$: $625ps - t_{sk2}$	437.5ps
$1 \to 2$: $350ps - t_{sk3} + t_{sk2}$	437.5ps
$2 \to 0$: $225ps + t_{sk3}$	325ps
$2 \to 1$: $525ps - t_{sk2} + t_{sk3}$	437.5 ps ✓

$T_{clk} \geq 437.5ps$

$t_{sk3} = 100ps$

$t_{sk2} = 187.5ps$

t_{sk2} = 187.5 ps.

t_{sk3} = 100 ps.

Tmin = 437.5 ps.

SID : solution

6) *Adders* (20 points, 20 minutes)

a) Ripple Carry Adder: Consider the following 4-bit ripple carry adder. The gates are from standard cell library and have the following properties:

$t_{and} = 4ps$, $t_{or} = 6ps$, $t_{xor} = 8ps$

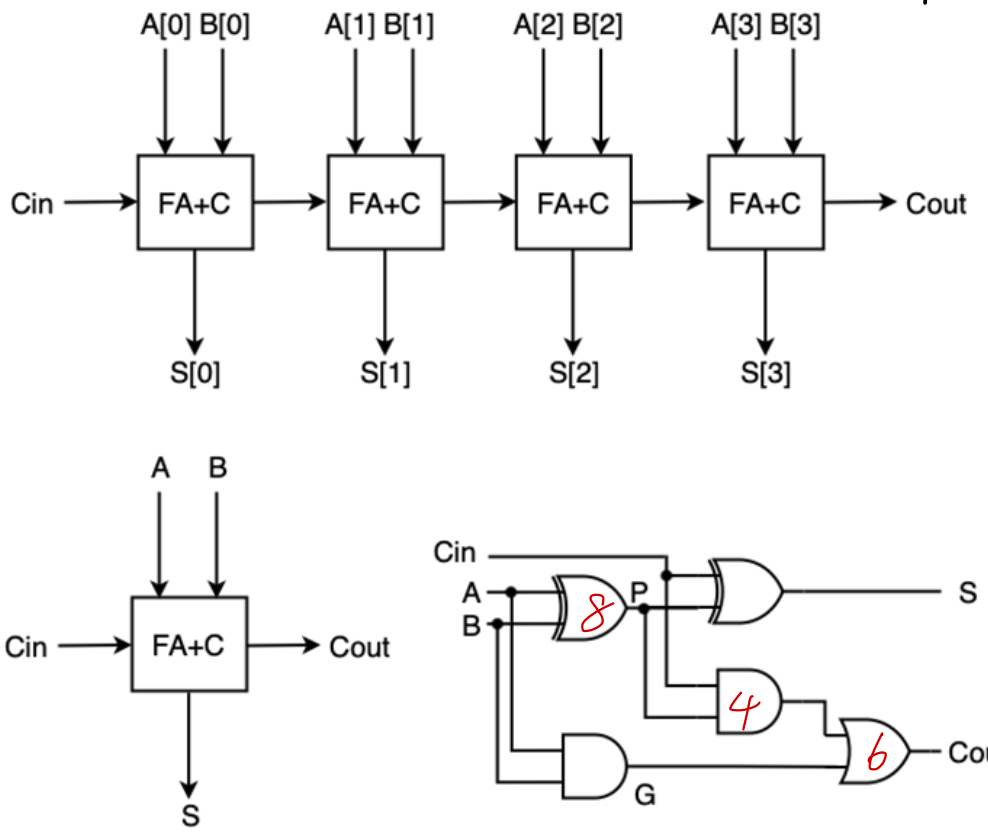


Figure 6.a.

Find the critical path of the circuit, and calculate the delay.

$A_0 \rightarrow Cout$: $\underbrace{8+4+6}_{A/B \rightarrow cout} + \underbrace{(4+6)}_{cin \rightarrow cout} \times 3 = 48ps$

$A_0 \rightarrow S_3$: $\underbrace{8+4+6}_{A/B \rightarrow cout} + \underbrace{(4+6)}_{Cin \rightarrow Cout} \times 2 + \underbrace{8}_{Cin \rightarrow S} = 46ps$

Critical Path: A[0]/B[0] Cout . (example: A[1] -> S[2])

Delay: 48 ps.

SID : solution

b) Carry-Lookahead Adder: Consider the following 4-bit carry-lookahead adder. The gates have the same delays as in part a). Again, find the critical path and calculate the corresponding path delay.

$t_{and} = 4ps$, $t_{or} = 6ps$, $t_{xor} = 8ps$

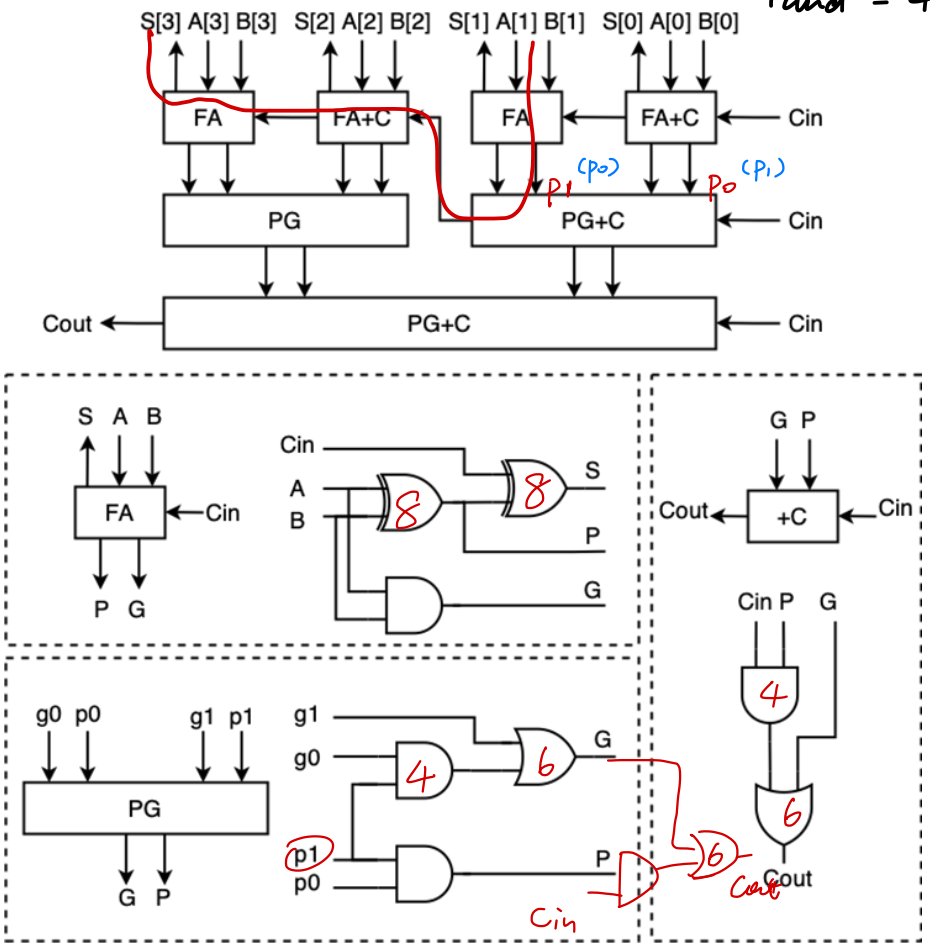


Figure 6.b.

$A_1/B_1 \rightarrow S_3$: $8 + 4 + 6 + 6 + 4 + 6 + 8 = 42ps$

(or $A_0/B_0 \rightarrow S_3$ if swapping P_0, P_1. Numbers are same)

8: FA, $A/B \rightarrow P$

$4 + 6 + 6$: PG+C, $P_1 \rightarrow$ cout

$4 + 6$: FA+C, cin → cout

8: FA, cin → S

Any A/B to Cout : $8 + 4 + 6 + 4 + 6 + 6 = 34ps$.

(P, G don't rely on Cin)

8: FA, $A/B \rightarrow P$

$4 + 6$: PG or PG+C, $P_1 \rightarrow G$

$4 + 6 + 6$: PG+C, $P_1 \rightarrow$ Cout.

Critical Path: ~~A[1]/B[1]~~ → S[3]

Delay: 42 ps.

SID : _____

7) *SRAM* (18 points, 20 minutes)

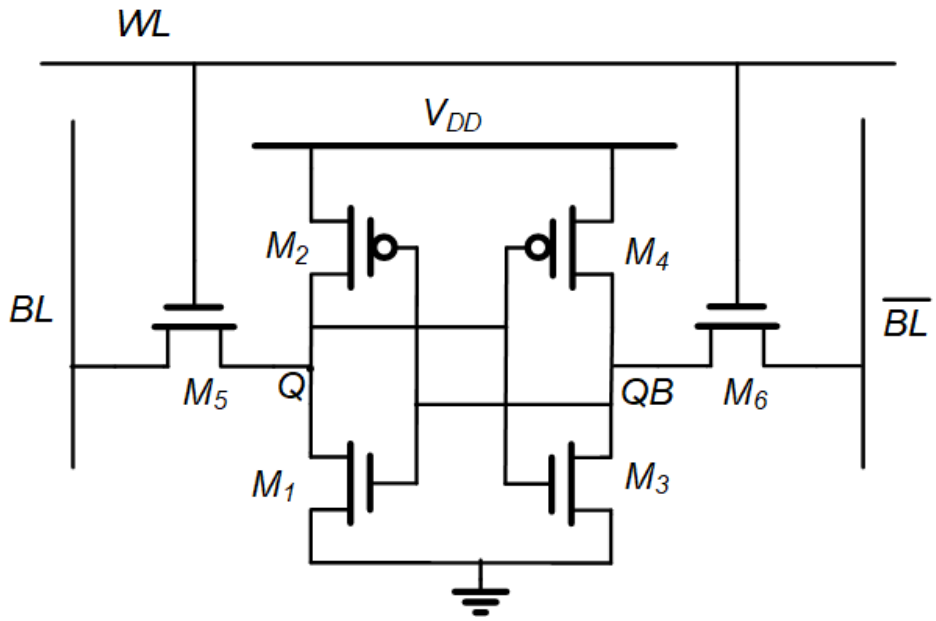


Figure 7

a) (3 pts) An SRAM cell in Figure 5 should be sized to be read stable and writable at nominal conditions. Size the SRAM cell by matching each transistor name with a choice of widths (W): A) 180nm, B) 135nm, C) 225nm. All transistors have equal length.

W_{M1}	C
W_{M2}	B
W_{M5}	A

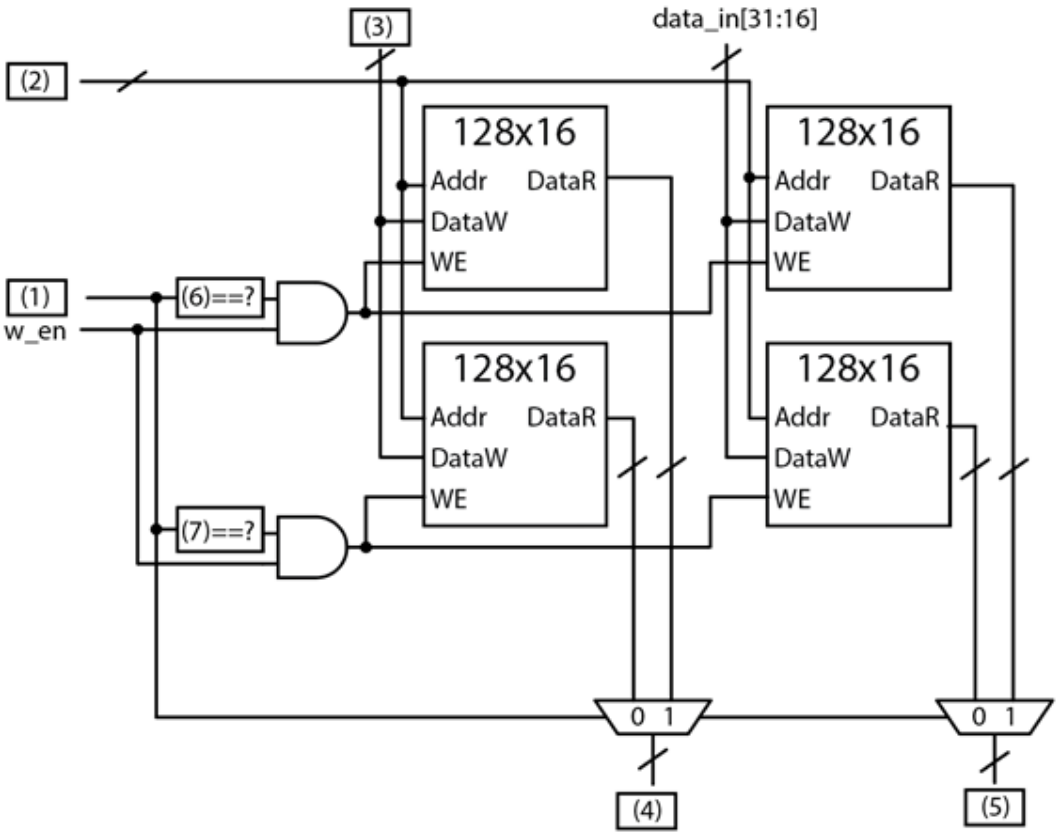
SID : _____

b) (8 pts) Please answer the following as true (T), false (F) or doesn't matter (X)

Increasing the supply voltage for the cross-coupled inverters (while keeping WL and BL at the nominal voltage) will hurt writability	T: Increasing the supply for the cross-coupled inverter increase the pull-up strength of PMOS It makes M5/6 harder to write
Decreasing the width of M5 and M6 will help the read speed	F: Decreasing the width of M5/6 reduce the strength and makes reading slower
Decreasing the width of M5 and M6 will help the read stability	T: Decreasing the width makes M5 and M6 weak So it's harder to override the existing value
Increasing the WL voltage will help writability	T: Makes M5/6 stronger, so easier to write in
Less cells per bitline speeds up writing	X: It's not relevant, speed up precharing doesn't count

SID : _____

c) Now you are given a memory block that is 128x16 bits. Suppose you would like to use multiple instances of this block to design a memory that is 256x32. The diagram below is a possible implementation. Fill the address signal names in the blanks and use the little-endian convention for address/data signals.



1	addr[7]
2	addr[6:0]
3	data_in[15:0]
4	data_out[15:0]
5	data_out[31:16]
6	1'b1
7	1'b0

SID : _____

8) Pipelining, *Energy* (13 points, 15 minutes)

We would like to examine some properties of a **4-stage** RISC-V datapath. The datapath has the following properties:

- Total capacitance of all gates: 10nF
- Max Frequency @ 1.0V: 400 MHz
- Supply voltage: 1.0V
- Register (flip-flop-based) overhead (sum of setup and clock-to-output delay) is 500ps (at 1.0V)

The CPU runs a workload with the following properties:

- Activity factor α: 0.1
- Number of instructions: 1000
- CPI (cycles per instruction) = 1

In this question, you may ignore leakage and short-circuit power.

a) (2 pts) What is the dynamic power consumption of this design?

$$P = \alpha C V_{DD}^2 f$$
$$= 0.1 \cdot 10nF \cdot 1^2 \cdot 400MHz$$

P = 400 mW.

b) (2 pts) How much energy is consumed by this workload?

$$E_{per\ cycle} = \alpha C V_{DD}^2 \cdot$$
$$= 0.1 \cdot 10nF \cdot 1^2$$

$$E_{workload} = E_{per\ cycle} \cdot instructions \cdot CPI$$
$$= E_{per\ cycle} \cdot 1000$$

E = 1 µJ.

SID : _____

c) (5 pts) If we increase the number of pipeline stages from 4 to 5, while perfectly rebalancing the logic, what would be the clock period of the design at the same 1-V supply?

$logic = \left(\frac{1}{400MHz} - 500ps\right) \cdot 4$ ← 4 stages

$= 2ns \cdot 4$ ↑ logic delay without register for 1 stage

$= 8ns$

$rebalanced = logic / 5 = 1.6ns$ ← register

$new\ critical\ path = rebalanced + 500ps$

$= 2.1ns$

T_{Clk} = 2.1 ns.

d) (4pts) We would like to reduce the power of this processor by lowering the supply voltage, so it operates at 400MHz. If the drain current is linearly proportional to supply voltage, what is the supply voltage that achieves this frequency?

$\frac{1}{2.1ns} = 476MHz$ ← current frequency

$\frac{400}{476} = \frac{new\ supply}{old\ supply}$

$\simeq 0.84$

V_{DD} = 0.84 V.