The University of Texas at Austin
EE460N/EE382N.1 Exam 1 (Fall 2017)

Mattan Erez, Sarbartha Banerjee, Madhuri Gontala, Gayatri Sivaraman

Your signature is your promise that you have **not cheated** and will not cheat on this exam, nor will you help others to cheat on this exam. Cheating means violating the explicit bullets below, or their spirit.

Signature:		Name:	Solution	EID:	

Note: Please be sure that your answers to all questions (and all supporting work that is required) **are contained in the box provided for each question;** we reserve the right to ignore anything else.

Note: **"I DON'T KNOW"** is a valid answer that automatically gives you 15% of the maximum possible grade. Be sure to either erase, or cross-out everything else and write I DON'T KNOW in, large, capital letters. A blank answer does not get you the 15%. No partial I DON'T KNOW credit (only on entire questions/parts that have a marked point value on the exam)

Note: Please put your name at the top of each page of the exam!

Note: For all questions, unless otherwise stated, find the most efficient solution. Efficiency counts.

Note: The exam is open book, open notes, bring anything type of exam, but there are constraints:

- **Do not ask anyone or anything for help. You may only search for text in PDFs or other material that you have downloaded.**
- **As a specific example of the above, no web searches.**
- **Do not use a calculator, phone, tablet, laptop, watch, … to compute anything. Not simple arithmetic and not complex simulations.**
- **Do not share any material related to the exam with anyone (or upload it anywhere). The exam will be made public soon.**
- **You should have no real need to type anything (or talk to a device), hence, significant typing is not allowed (minor Ctrl+F searches are OK).**

The exam has a total of 3 questions and 5 pages. It's a good idea to read through first some questions might be easier for you than others.

For each page that doesn't include your name and EID you may be deducted 1 point!

Name: EID:

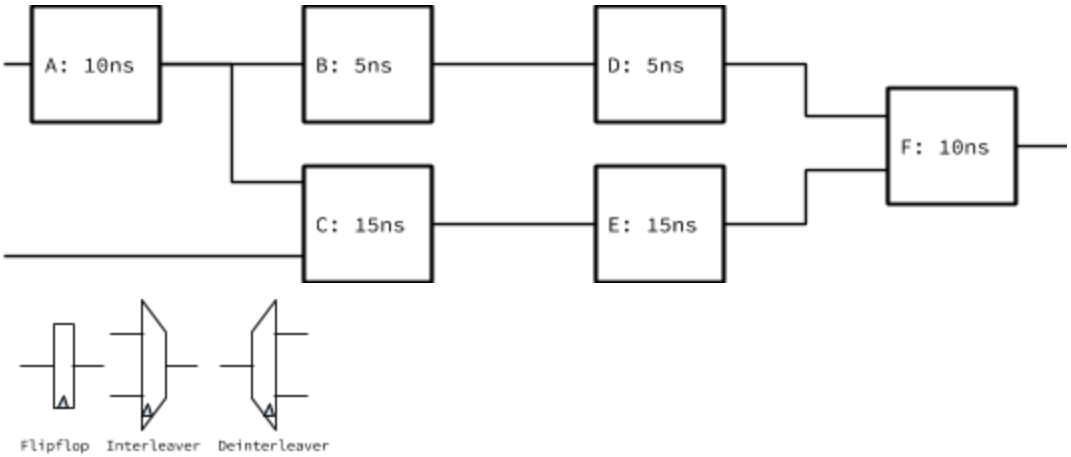
Question 1 ():

Complete the table below, where each row represents a different system. If a cache function is invalid explain why and don't bother filling in the rest. All caches use random replacement. All memory addresses are 16 bits and all memory operations are byte addressable.
WB/WT==writeback or writethrough. The ^ symbol is an XOR operation.

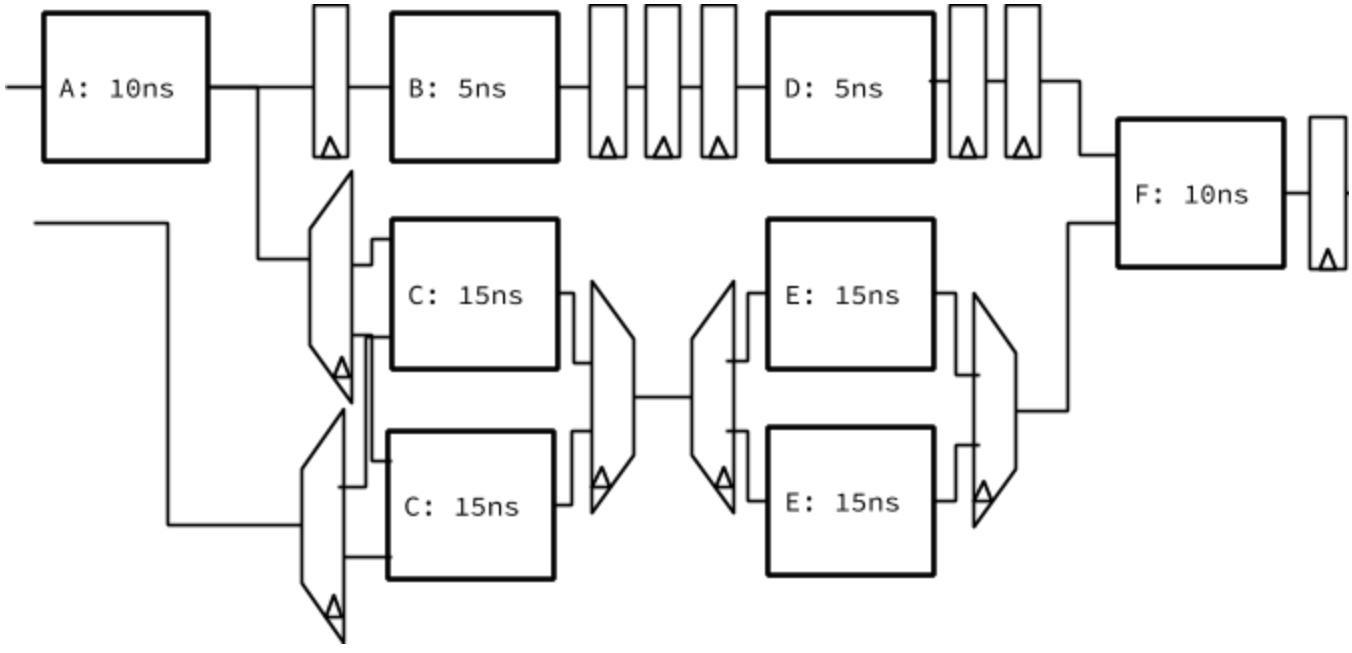
Cache function	Valid?	Block size(B)	# sets	# ways	Cache capacity (B)	WB or WT?	Tag arrays size (bits)
offset=addr[2..0] idx=addr[6..3] tag=addr[15..7]	**Yes**	**8**	**16**	4	**512**	WT	**640**
offset=addr[0] idx=addr[7..1] tag=addr[12..8]	**No. Some address bits not used to determine cache hit/miss**	2		2	512	WB	
offset=addr[1..0] idx=addr[9..2] tag=addr[15..10]	**Yes**	**4**	**256**	**1**	1,024	**WB**	2,048
offset=N/A idx=addr[7..0] tag=addr[15..8]	**Yes**	1	**256**	**2**	**512**	**WB**	5,120
offset=addr[1..0] idx=addr[3..2] ^ addr[1..0] tag=addr[15..4]	**No. A "single" cache block is spread across multiple sets. Can't make this work**			1		WB	56
offset=addr[1..0] idx=addr[6..2] ^ addr[15..11] tag=addr[15..7]	**Yes**	4	32	2	**256**	WT	**640**

Question 2 (points total):

Pipeline the circuit below. Optimize throughput (inputs processed per ns) and cost ($). You may add any number of the blocks in the circuit (A-F), edge-triggered flipflops, edge-triggered interleavers, and edge-triggered de-interleavers (see figure below). No other modifications are permitted. The latency of flipflops, interleavers, and de-interleavers is 0ns. The cost of any item (whether already there or one you add), regardless of which item is $1. You have a total budget of $25 and the components already drawn below cost $6. Draw your answer on top of the diagram below if at all possible.



There is possibly more than a single correct solution. Things to think about: (a) pipeline needs to be well-formed with (i) same number of latches on all paths, and all signals "aligned" w.r.t. unpipelined version (i.e., an interleaver doesn't reduce the delay of a circuit blow); (b) you need to interleave to get the best performance/cost (though no interleaving is not far); (c) multiple inputs should be handled correctly when interleaving; (d) if not treating the interleavers as also latching the inputs, then need a latch for each input signal on each path; (e) interleaving 3 ways is a bit tricky (and exceeds budget if done correctly); and (f) being consistent in assumptions and solution is important even if something in the assumptions is off.



Name: EID:

Question 3 (points total):

Consider the LC3b microarchitecture diagram (attached to the exam), where the memory system includes an instruction cache, a data cache, and DRAM. You are asked to evaluate two options for improving the performance of the LC3b microarchitecture on "typical" programs. Both options have the same impact on complexity, cycle-time, cost, …

Option 1: Merge State 35 and State 32, such that together they require just a single cycle.

Option 2: Improve the data cache hit rate from 0.9 to 0.95 on typical programs.

Assume:

- Each state requires one cycle to complete
- There is a perfect instruction cache (i.e., State 33 always completes in a single cycle and an instruction fetch never accesses DRAM)
- There is a data cache that on typical programs has a 90% hit rate and a 1-cycle latency
- There is no second-level cache
- DRAM requires 100 cycles if the memory address accessed is not in the cache
- In a typical program there are 10 instructions for every control-flow instruction (a control-flow instruction is one that changes the PC explicitly)
- The different control flow are executed about the same number of times in a typical program (i.e., BR, JSR, JMP, and RET instructions are spread evenly throughout execution)
- The TRAP instruction is never called

What parameter is not specified? Explain why this particular parameter is crucial:

Fraction of instructions that are memory instructions. Need to know how much time will be saved by increasing hit rate.

Express the tradeoff as a speedup **equation** of Option 2 compared to Option 1 (you can round a little):

$$Speedup = \frac{Time\ option\ 1}{Time\ option\ 2} = \frac{N\cdot 4+N\cdot f\cdot((0.9\cdot 1+0.1\cdot 100)+1)+N\cdot 0.1\cdot 0.5\cdot 1}{N\cdot 5+N\cdot f\cdot((0.95\cdot 1+0.05\cdot 100)+1)+N\cdot 0.1\cdot 0.5\cdot 1} > 1$$

$$\frac{4.05+11.9\cdot f}{5.05+6.95\cdot f} > 1$$

$$11.9\cdot f > 6.95\cdot f+1$$

$$f > \frac{1}{5}$$

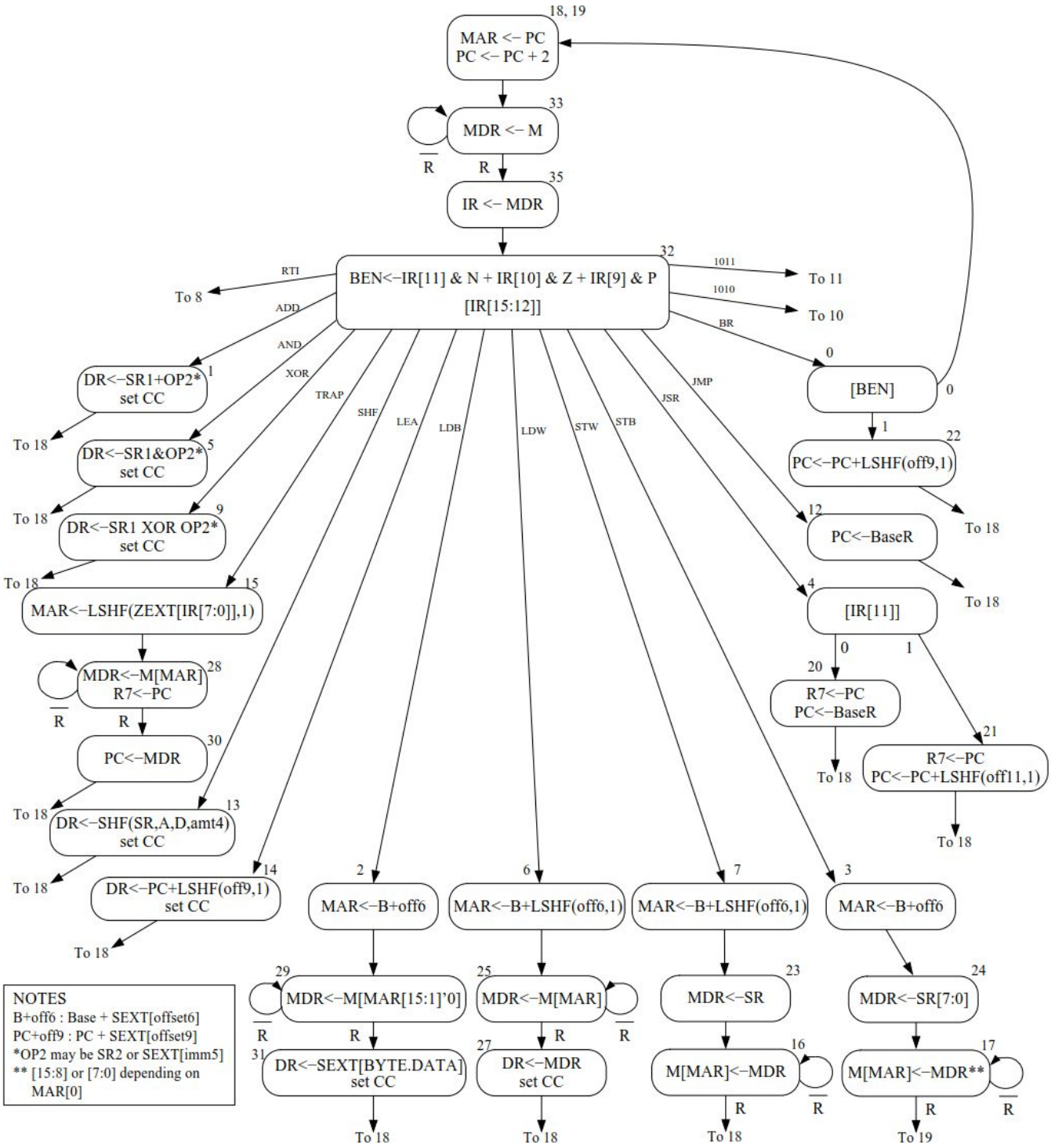
At what value of the missing parameter will Option 2 be better than Option 1 (you can round a little)?

1/5

Is the parameter likely to have this value for a typical program? Explain your answer.

Seems reasonable considering we have few registers and will likely access memory fairly frequently.

Name: EID:


18, 19
MAR <– PC
PC <– PC + 2
33
MDR <– M
R
R
35
IR <– MDR
32
BEN<–IR[11] & N + IR[10] & Z + IR[9] & P
[IR[15:12]]
RTI
To 8
ADD
AND
XOR
TRAP
SHF
LEA
LDB
LDW
STW
STB
JSR
JMP
1011
To 11
1010
To 10
BR
0
[BEN]
0
1
22
PC<–PC+LSHF(off9,1)
To 18
12
PC<–BaseR
To 18
4
[IR[11]]
0
1
20
R7<–PC
PC<–BaseR
To 18
21
R7<–PC
PC<–PC+LSHF(off11,1)
To 18
1
DR<–SR1+OP2*
set CC
To 18
5
DR<–SR1&OP2*
set CC
To 18
9
DR<–SR1 XOR OP2*
set CC
To 18
15
MAR<–LSHF(ZEXT[IR[7:0]],1)
28
MDR<–M[MAR]
R7<–PC
R
R
30
PC<–MDR
To 18
13
DR<–SHF(SR,A,D,amt4)
set CC
To 18
14
DR<–PC+LSHF(off9,1)
set CC
To 18
2
MAR<–B+off6
6
MAR<–B+LSHF(off6,1)
7
MAR<–B+LSHF(off6,1)
3
MAR<–B+off6
29
MDR<–M[MAR[15:1]'0]
R
R
31
DR<–SEXT[BYTE.DATA]
set CC
To 18
25
MDR<–M[MAR]
R
R
27
DR<–MDR
set CC
To 18
23
MDR<–SR
16
M[MAR]<–MDR
R
R
To 18
24
MDR<–SR[7:0]
17
M[MAR]<–MDR**
R
R
To 19
NOTES
B+off6 : Base + SEXT[offset6]
PC+off9 : PC + SEXT[offset9]
*OP2 may be SR2 or SEXT[imm5]
** [15:8] or [7:0] depending on
MAR[0]