Name:_____

Problem 1 (30 points):

Part a (5 points): A bus has Data, Address, and Control lines. MSYN and SSYN are **control** lines. A bus that has these lines are (circle one) synch/(**asynch**). Why is that the case?

MSYN and SSYN are required for handshaking instead of the clock

Part b (5 points): We wish to perform residue arithmetic using moduli 4, 7, 9. How many numbers can we represent uniquely in the residue domain?

252

What is the value of the decimal number 249 in the residue domain?

146

Part c (5 points): Single decimal digit, saturating arithmetic

3 + 5 = **8**

3 - 5 = **0**

Part d (5 points): Pipelining increases the performance of a processor if the pipeline can be kept full with useful instructions. Two reasons that often prevent the pipeline from staying full with useful instructions are (in two words each):

Conditional Branches

Flow Dependencies

Name:_____

Problem 1 continued:

Part e (5 points):

```
for (i=1; i < 500; i++)
    A[i] = (B[i] + A[i-1])/4;
```

Is the loop vectorizable? Circle one: yes/(**no**) Explain why/why not (10 words should be enough).

Each iteration is dependent on the previous iteration

Part f (5 points): We wish to represent the decimal value 27 with an 8-bit floating point number, having a sign bit, 4 bits of excess-7 exponent and three bits of fraction. The exact representation is

0 1011 1011

which will not fit in the available bits. So, we do some rounding. In the IEEE Floating Point standard, there are four rounding modes. Assuming the above format is legitimate, what exact decimal value would I get using the unbiased nearest rounding mode? Show the representation.

Decimal Value	2 8
8−bit representation	0 1 0 1 1 1 1 0

Name:_____

Problem 2 (20 points):

Part a. A five instruction sequence executes according to Tomasulo's algorithm. Each instruction is of the form ADD DR,SR1,SR2 or MUL DR,SR1,SR2. ADDs are pipelined and take 9 cycles (F-D-E1-E2-E3-E4-E5-E6-WB). MULs are also pipelined and take 11 cycles (two extra execute stages). The microengine must wait until a result is in a register before it sources it (reads it as a source operand).

The register file before and after the sequence are shown below (tags for "After" are ignored).

Before

	V	tag	value
R0	1	z	4
R1	1	z	5
R2	1	z	6
R3	1	z	7
R4	1	z	8
R5	1	z	9
R6	1	z	10
R7	1	z	11

After

	V	tag	value
R0	1		310
R1	1		5
R2	1		410
R3	1		31
R4	1		8
R5	1		9
R6	1		10
R7	1		21

Complete the five instruction sequence in program order in the space below. Note that we have helped you by giving you the opcode and two source operand addresses for instruction 4. (The program sequence is unique.)

1	ADD R7, R6, R7
2	ADD R3, R6, R7
3	MUL R0, R3, R6
4	MUL R2, R6, R6
5	ADD R2, R0, R2

Part b. In cycle 1 instruction 1 is fetched. In cycle 2, instruction 1 is decoded and instruction 2 is fetched. In cycle 3, instruction 1 starts execution, instruction 2 is decoded, and instruction 3 is fetched.
Assume the reservation stations are all initially empty. Put each instruction into the next available reservation station. For example, the first ADD goes into "a". The first MUL goes into "x". Instructions remain in the reservation stations until they are completed.Show the state of the reservation stations at the end of cycle 8.

Note: To make it easier for the grader, when allocating source registers to reservation stations, please always have the higher numbered register be assigned to SR2.

1	z	10	1	z	11	a
1	z	10	0	a	11	b
0	x	4	0	y	6	c

+

0	b	7	1	z	10	x
1	z	10	1	z	10	y
						z

x

Name:_____

Problem 2 continued:

Part c. Show the state of the Register Alias Table (V, tag, Value) at the end of cycle 8.

	V	tag	value
R0	0	x	4
R1	1	z	5
R2	0	c	6
R3	0	b	7
R4	1	z	8
R5	1	z	9
R6	1	z	10
R7	0	a	11

Name:_____

Problem 3 (20 points):

A processor with 32 MB byte-addressable physical memory has a virtually-indexed, physically-tagged write-back cache. The cache is 4-way set associative and 2 bits are used for each block to implement a victim/next-victim pseudo-LRU replacement policy. The size of the tag store of the cache is 9216 bits. The cache can store 512 blocks of 128 bytes of data. Based on this information, answer the following questions:

Part a. How many frames are there in physical memory?

9216/512 = 18 bits of tag store entry for each block
18 = (V (1 bit) + M (1 bit) + Victim/Next-victim (2 bits) + Tag (x bits)
x = 14 bits
PFN is 14 bits
2^{14} **frames**

Part b. What is the page size?

32 MB = 2^{25}
Physical address is 25 bits
14 of these bits are used for the PFN
25 - 14 = 11 are used for page offset
2^{11} **bytes (2KB)**

Part c. How many bits of the cache index come from the ”virtual page number”?

128-byte blocks: 7 bits used for identifying the byte in block
128 entries in the cache: 7 bits used for index
(7 + 7) - 11 = 3 bits come from virtual page number
3

Part d. What is the size of the data store in bits?

512 x 128 x 8 = 2^{19}
2^{19} **(524288) bits**

Name:_____

Problem 4 (15 points):

Suppose we have a 13-bit floating point format in which 54.5 is represented as 0100010110100.

Part a. What is the precision of this format?

54.5 = 110110.1 = 1.101101 x 2^5
1 bit for sign, 4 bits for exponent, 8 bits for fraction
bias = 3
precision = 1 + 8 = 9 bits

Part b. What is the range of normalized positive numbers?

smallest number : 1.0 x 2^{-2} = 0.25
largest number : 1.11111111 x 2^{11} = 4088

Part c. What is the smallest positive subnormal number representable in this format?

0.00000001 x 2^{-2} = 2^{-10} = 1/1024

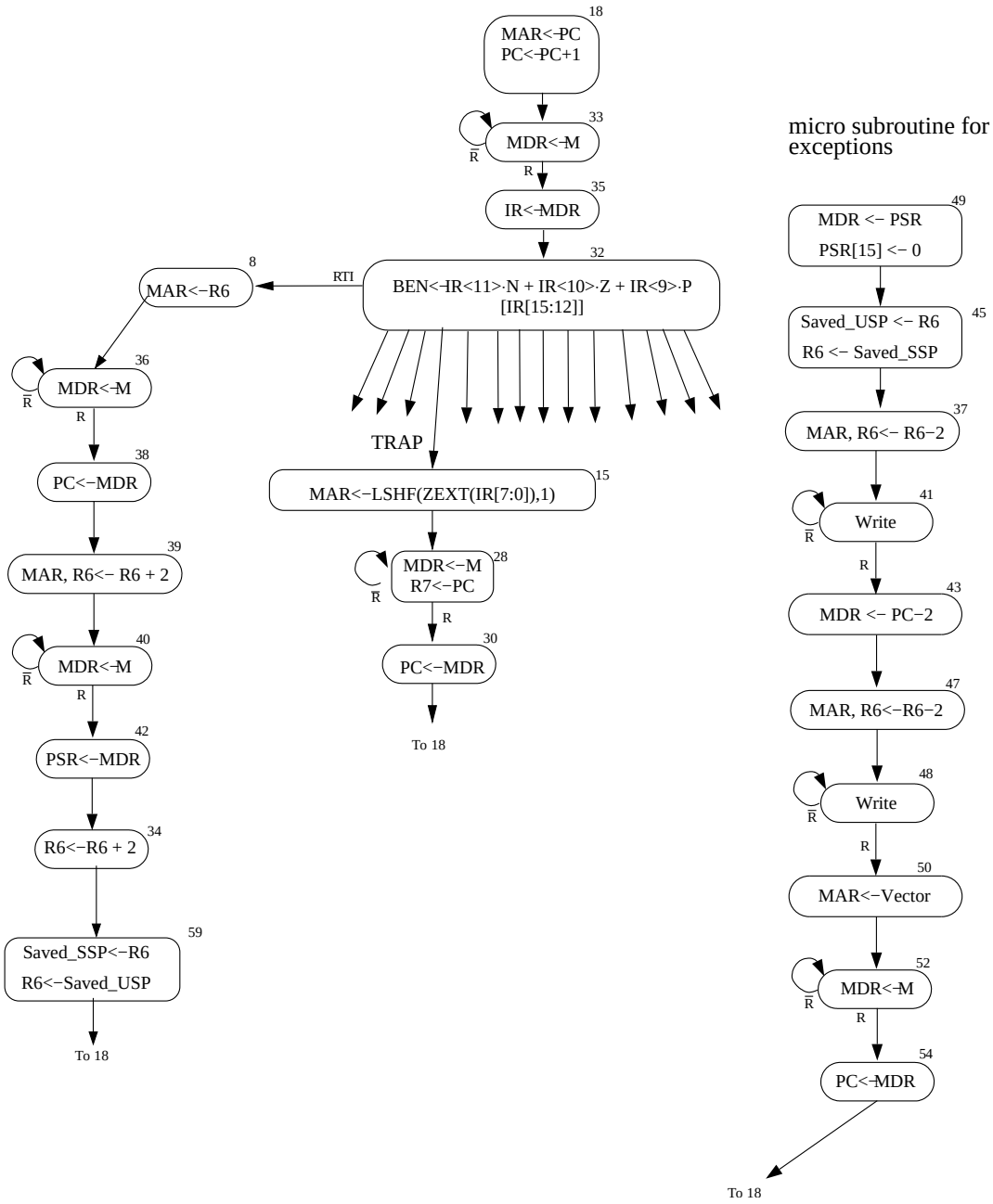
Name:_____

Problem 5 (15 points):

In Lab Assignment 3 we augmented the datapath, the state machine and the microsequencer in order to handle interrupts, page faults and access control violations.

In this problem, we are interested in extending your machine to take an exception in case a TRAP instruction produces an illegal starting address. If the address obtained from the trap vector table is odd or it maps to page 0 (address range [0:511]), your machine should take an exception. We will call this exception an "illegal service routine address" exception and assign it an exception vector 4.

Shown below is a typical state machine from Lab 3. It has states for the RTI instruction, the TRAP instruction and the micro subroutine for exception handling.



Name:_____

Problem 5 continued:

Part a. Specify the logic function to test an illegal service routine address. (3 points)

ISRA = MDR[0] + NOT(MDR[15] + MDR[14] + MDR[13] + MDR[12] + MDR[11] + MDR[10] + MDR[9])

Part b. Modify the given state machine to test the "illegal service routine address" exception. Specify clearly the operations performed in each state. Show clearly how the control is transferred to the micro subroutine in case of an exception. Your solution should cause only such changes to the datapath as are absolutely necessary. You may not need to use all the states provided below. (12 points)