

READ THIS FIRST:

We recommend first trying this assignment in a single sitting. The midterm exam time period is 80 minutes long. Find a quiet place, grab your cheat sheet* and a pencil, and set a timer. Once you're done, set it aside for a bit. Come back later and look over your answers, trying to find errors and treating this more like a regular homework.

We will not finish covering ADCs/DACs until Wednesday, October 21st. We've added a blank page before the ADC/DAC questions. It would be a good idea to rip off the last two pages and try the rest as a test-exam before the 21st. Estimate ~15 minutes for the ADC/DAC questions (set a 65 minute timer). We recognize that this deadline turnaround is a little quick, but we would like to have this assignment graded before lecture on Monday the 26th so that you can ask questions.

Finally, while this is a practice midterm, it does not necessarily represent the exact material on the midterm. You are responsible for knowing all topics covered in this course. For example, the astute student may notice that the material from Week 3 of lectures is conspicuously absent. You are still responsible for knowing the material from Week 3...

*One physical piece of 8.5x11" paper (you may use both sides). Notes must be handwritten.

Name: _____

username: _____

EECS 373 Practice Midterm / Homework #5

Fall 2015

Due 11:55PM Friday, October 23rd

Sign the honor code:

I have neither given nor received aid on this exam nor observed anyone else doing so.

Problem #	Points
1	/10
2	/10
3	/10
4	/20
5	/5
6	/10
7	/10
8	/10
9	/15
Total	/100

NOTES:

- **PUT YOUR NAME/UNIQUENAME ON EVERY PAGE TO ENSURE CREDIT!**
- Can refer to the ARM Assembly Quick Ref. Guide and 1 page front/back cheat sheet
- Can use a basic/scientific calculator (but not a phone, PDA, or computer)
- Don't spend too much time on any one problem.
- You have 80 minutes for the exam.
- The exam is XX pages long, including the cover sheet.
- Show your work and explain what you are doing. Partial credit w/o this is rare.

Name: _____

username: _____

1) **Fill-in-the-blank or circle the best answer. [10 pts, all or no points for each question]**

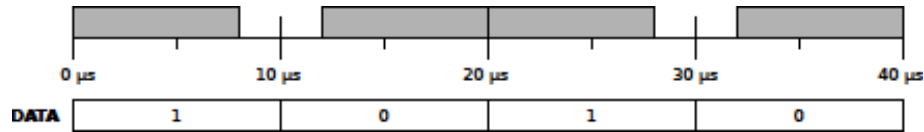
- a) ARM EABI-compliant functions can pass up to _____ `uint32_t` arguments in registers.
- b) When fetching a new instruction from memory, a Cortex-M will ***fetch the instruction in parallel with a load/store operation / wait for any load/store operations to finish.***
- c) A PWM controller with a 40 MHz clock and 40% duty cycle generates _____ ns pulses.
- d) An ARM EABI-compliant procedure that calls another procedure should ***always / sometimes / never*** save and restore the **`lr`** register.
- e) Generally, a ***SPI / I2C*** bus will have a faster maximum clock speed.
- f) You are building a new platform, the Mini-duino for hackers. To make your expansion header as small as possible while staying as flexible as possible, you should probably pin out only ***UART / SPI / I2C.***
- g) In the Verilog hardware description language, an @ (_____) block would be used when implementing a flip-flop.
- h) An executing interrupt handler ***can never be interrupted / might be interrupted / is always interrupted*** by a newly arriving interrupt.
- i) If multiple devices share a single interrupt line and generate interrupts at the same rate, then processor workload ***remains constant / grows linearly / grows quadratically*** with the number of devices.
- j) The _____ is the very first thing found in memory.

Name: _____

uniqname: _____

2) Logic Design. [10]

Design a circuit for transmitting data for visible light communication. Because the data is transmitted over a visible light medium, it must be *DC balanced*, that is the amount of time the LED is on or off must be independent of the number of 0's and 1's in the data stream. Your circuit should encode data using Variable Pulse Position Modulation (VPPM). Transmitting a 0 should turn the LED off for 2 μs and on for 8 μs . Transmitting a 1 should turn the LED on for 8 μs and off for 2 μs . As example:



Your circuit should take an input clock of 10 MHz clock (CLKIN) and a 100 kHz binary data stream (DATA) and output a signal LED_OUT. You may *not* assume that DATA and CLKIN are from the same clock domain. You may use synchronously resettable D flip-flops and n -bit binary counters (make sure to specify the value of n). You may express combinational logic as Boolean expressions or using standard logic gates. Label things and write neatly.

DATA

LED_OUT

CLKIN

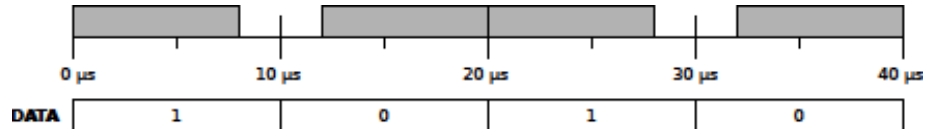
Name: _____

uniqname: _____

3) Hardware Design. [10]

Write a Verilog module that implements the same VPPM encoding as question (2). For convenience, the timing diagram and explanation are repeated here. They are the same.

Design a circuit for transmitting data for visible light communication. Because the data is transmitted over a visible light medium, it must be *DC balanced*, that is the amount of time the LED is on or off must be independent of the number of 0's and 1's in the data stream. Your circuit should encode data using Variable Pulse Position Modulation (VPPM). Transmitting a 0 should turn the LED off for 2 μ s and on for 8 μ s. Transmitting a 1 should turn the LED on for 8 μ s and off for 2 μ s. As example:



CLKIN is the 10 MHz core clock. DATA is a 100 kHz binary data stream that does *not* necessarily come from the same clock domain as CLKIN. Your Verilog code must be synthesizable.

```
module vppm(input clk_in, input data, output led_out)
```

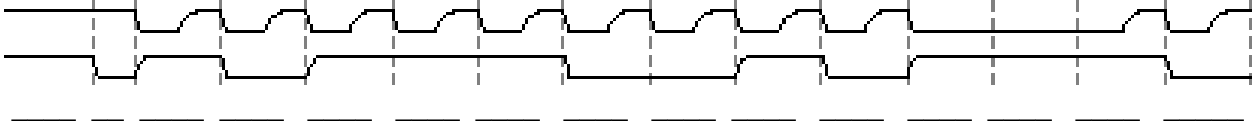
```
endmodule
```

Name: _____

username: _____

4) Serial Buses and Debugging. [20]

a.) A student probes two wires on a mystery board and captures the following waveform:



- I.) What is the student probing? _____
- II.) Label the two wires.
- III.) The waveform is broken down into regions. Add a label to each region describing what is happening in that region. (fill in the blanks below the waveform)
- IV.) Describe in words what this waveform is doing.

b.) The student tried to copy this board, building their own version. When they powered it up, however, it didn't work. Probing the board, the student saw this waveform:



- I.) The student double-checked their board and every wire is connected. Give a possible explanation for what is wrong with this board.
- II.) How could you fix this problem?

c.) The student gave it another try, starting with a completely fresh design. When they tried the new design, it still didn't work. Probing the board, the student saw this waveform:



- I.) The student double-checked their board and every wire is connected. Give a possible explanation for what is wrong with this board.
- II.) How could you fix this problem?

Name: _____

username: _____

- 5) **ARM Addressing Modes.** Assume that **r3=0xaabbccdd** and **r1=0x00003000**, and all other registers and memory are initialized to zero. After running the following code, what are the values of registers **r1**, **r3**, and **r5**? Annotate each line of code to show your partial results including any effective addresses and intermediate values in the registers and memory. **[5 pts]**

```
str  r3, [r1, #1]
ldrb r5, [r1], #2
orr  r5, r5, #0x05
strh r3, [r1, #-4]!
ldr  r3, [r1]
```

r1 = _____

r3 = _____

r5 = _____

Name: _____

username: _____

6) Arm Assembly Language [10]

Encoding T1 ARMv7-M
MOVT<c> <Rd>, #<imm16>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	0	i	1	0	1	1	0	0	imm4	0	imm3	Rd	imm8															

d = UInt(Rd); imm16 = imm4:i:imm3:imm8;
if d IN {13,15} then UNPREDICTABLE;

Encoding T1 All versions of the Thumb instruction set.
MOVS <Rd>, #<imm8>
MOV<c> <Rd>, #<imm8>

Outside IT block.
Inside IT block.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	Rd	imm8									

d = UInt(Rd); setFlags = !InITBlock(); imm32 = ZeroExtend(imm8, 32); carry = APSR.C; (shift_t, shift_n) = (SRTYPE_LSL, 0);

Encoding T1 All versions of the Thumb instruction set.
STR<c> <Rt>, [<Rn>, <Rm>]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	0	0	0	Rm	Rn	Rt						

t = UInt(Rt); n = UInt(Rn); m = UInt(Rm);
index = TRUE; add = TRUE; wback = FALSE;
(shift_t, shift_n) = (SRTYPE_LSL, 0);

a) What does the instruction 0x5088 do when executed?

b) Fill in machine code in hex around 0x5088 such that the value 22 is written to address 0x20000008.

0x5088

Name: _____

unique: _____

7) NVIC and Memory Map Comprehension. [10 pts]

Write a C function `void clear_interrupt(int x)` that clears pending interrupt `x`. You need not check to validate that `x` is a legal interrupt number. The table below might be useful.

Table 6-1 NVIC registers

Address	Name	Type	Reset	Description
0xE000E004	ICTR	RO	-	<i>Interrupt Controller Type Register, ICTR</i>
0xE000E100 - 0xE000E11C	NVIC_ISER0 - NVIC_ISER7	RW	0x00000000	Interrupt Set-Enable Registers
0xE000E180 - 0xE000E19C	NVIC_ICER0 - NVIC_ICER7	RW	0x00000000	Interrupt Clear-Enable Registers
0xE000E200 - 0xE000E21C	NVIC_ISPR0 - NVIC_ISPR7	RW	0x00000000	Interrupt Set-Pending Registers
0xE000E280 - 0xE000E29C	NVIC_ICPR0 - NVIC_ICPR7	RW	0x00000000	Interrupt Clear-Pending Registers
0xE000E300 - 0xE000E31C	NVIC_IABR0 - NVIC_IABR7	RO	0x00000000	Interrupt Active Bit Register
0xE000E400 - 0xE000E4EC	NVIC_IPR0 - NVIC_IPR59	RW	0x00000000	Interrupt Priority Register

```
void clear_interrupt(int x) {
```

```
}
```

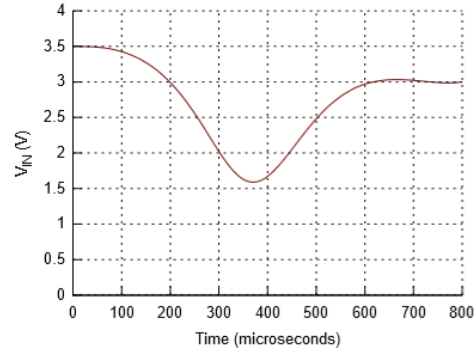
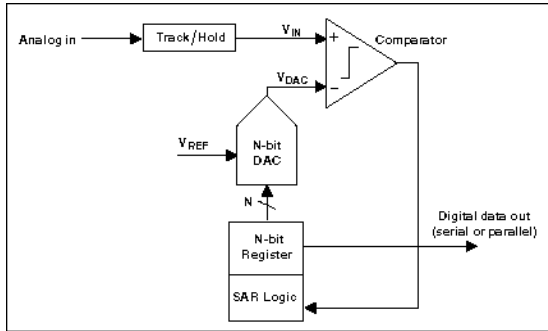
Name: _____

username: _____

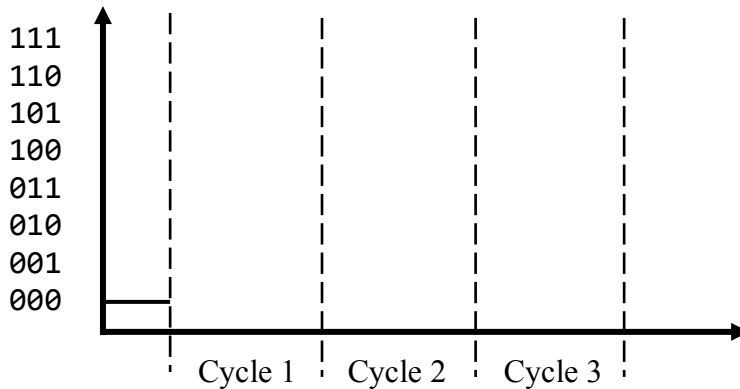
This page intentionally left blank. Gap page before ADC / DAC questions.

8) Data Converters #1 [10 pts]

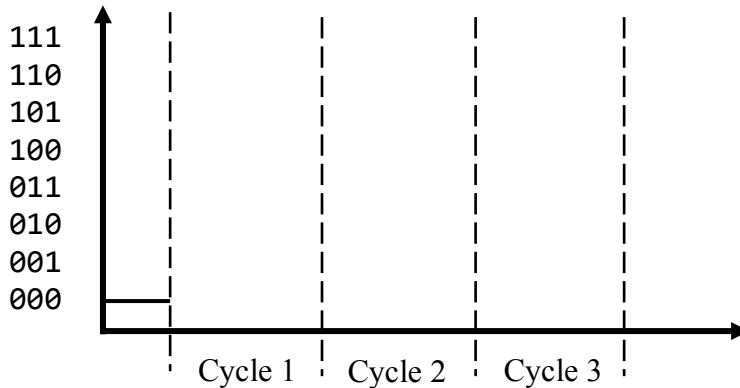
Assume you have a 3-bit successive approximation register (SAR) ADC. V_{REF} is 4 V and each cycle takes 100 μ s. The signal V_{IN} is changing over time, as shown in the graph. Recall that the “Track and Hold” block will latch the input value at the beginning of a measurement and the ADC will use the held value during every cycle. Show how the SAR would approximate the analog input over three cycles. Label the cycles on the x-axis and show the approximation as a meandering stair-step line on the graph. Assume that the N-bit DAC uses a voltage divider with only a single resistance.



a.) Plot the internal DAC input if the ADC begins sampling at 100 μ s:



b.) Plot the internal DAC input if the ADC begins sampling at 100 μ s but DOES NOT use track and hold (the signal is connected directly to the comparator input):



c.) This ADC does not capture the dip in the supplied waveform very well. Your boss suggests improving the ADC by changing it from a 3-bit to a 4-bit SAR (changing nothing else). Will the new ADC capture this signal better or worse, why?

Name: _____

uniqname: _____

9) Data Converters #2 [15 pts]

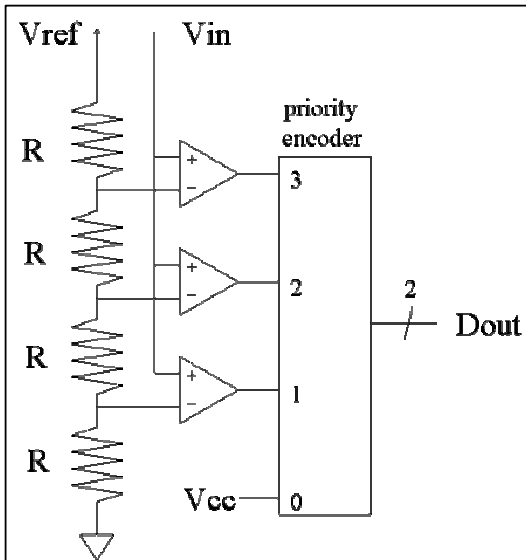
Consider the ADC and DAC found below. Assume V_{ref} is 20 V for the ADC and 16 V for the DAC and that both converters have an absolute error of up to $\pm 1/4$ LSB. The output of the ADC (Dout) is connected to the input of the DAC (Din).

If 9 V is supplied as the input to the ADC (V_{in}), what are all of the possible values you might get on the output of the DAC (V_{out}): _____.

Label every node and wire in the circuit with its value (analog or “digital”).

Show your work. [15 pts]

ADC



DAC

