

**HOMEWORK ASSIGNMENT #1 - SOLUTION**Due Wednesday, February 4<sup>th</sup>, 2009

*Be sure to put your homework **team number** and all members' **names** on the first page of your homework submission.*

**1. (10 points) Memory Structures**

A. For each of the below memory structures, express the memory size in bytes using the appropriate prefix (K, M, G, etc).

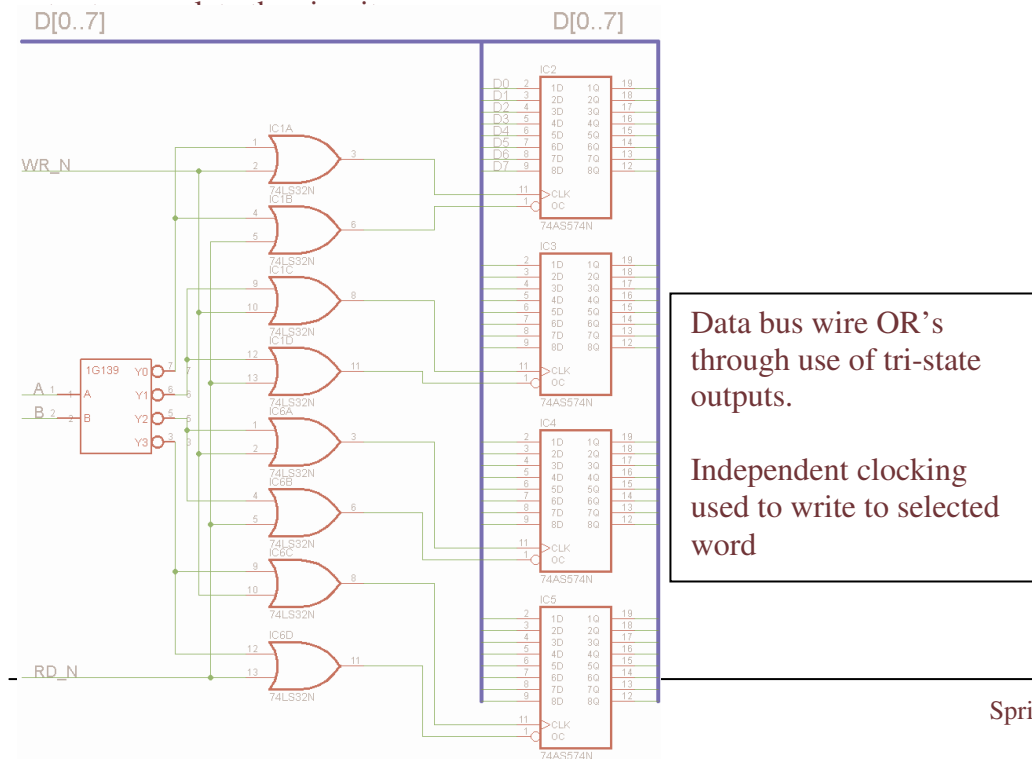
- a. 4096 x 4b → 2kB  
 b. 8192 x 12b → 12kB  
 c. 2097152 x 32b → 8MB  
 d. 8589934592 x 64b → 64GB

B. For byte-wide memories of sizes 1.5kB, 1.5MB, 1GB, and 1TB, what is the number of memory locations and the range of addresses for each device, all expressed in hexadecimal. How many address bits does each device require?

Memory Size	# of locations (Hex)	Address range:	#address bits
1.5kB	0x600	0x000 to 0x5FF	11
1.5MB	0x180000	0x000000 to 0x17FFFF	21
1GB	0x40000000	0x00000000 to 0x3FFFFFFF	30
1TB	0x1000000000	0x000000000 to 0xFFFFFFFF	40

**2. (15 points) Memory Construction**

Use SN74LVTH574 and SN74LVC1G139 devices to create a 4 x 8b SRAM. Obtain datasheets for them from Texas Instruments, Inc. Your circuit should have the following external connections; A[1:0], D[7:0], /WR, /RD. You may draw the schematic by hand (neatly), or use one of the many CAD programs available. Use the minimum number of devices and additional



### 3. (15 points) Memory Operation

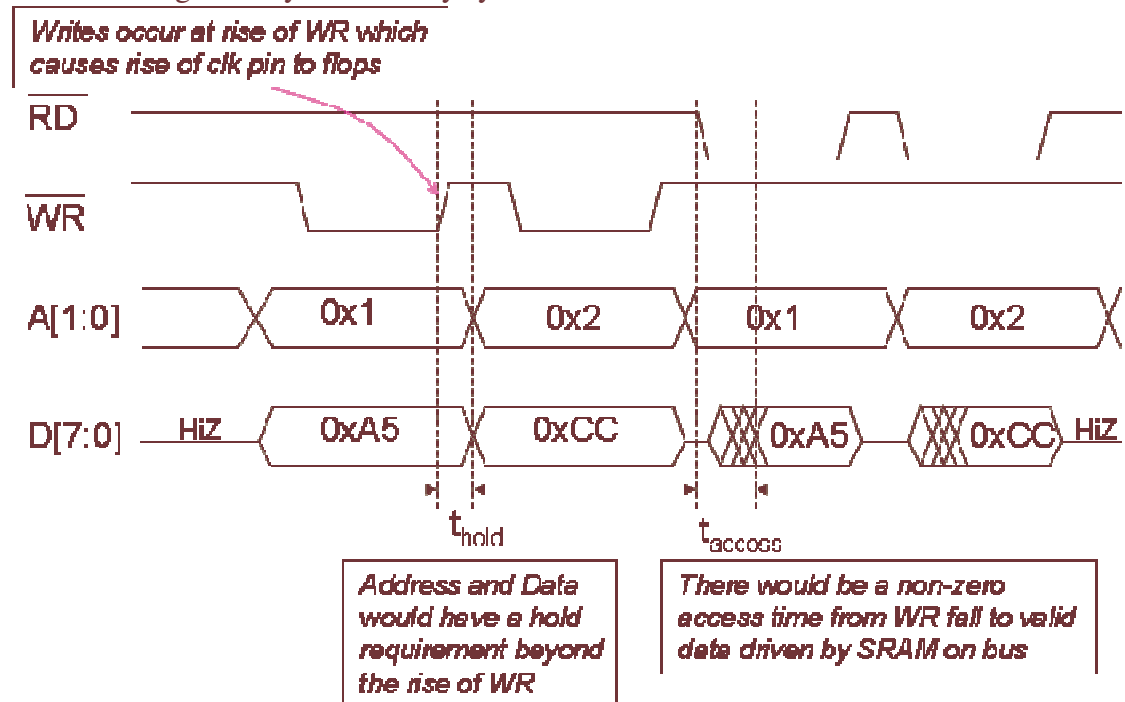
A. For the circuit you designed in #2 above, draw the waveforms ( $\overline{WR}$ ,  $\overline{RD}$ , A[1:0], D[7:0]) that would be required to perform the following sequence of operations.

- 1) write the value 0xA5 into the register at address 1
- 2) write the value 0xCC into the register at address 2
- 3) read the value in the register at address 1
- 4) read the value in the register at address 2

Ensure that any expected or required signal delays are made obvious on your diagram. Annotate your drawing to clearly indicate what is happening.

Draw vectors (multi-bit signals like A[1:0] & D[7:0]) as single busses (not bit blasted).

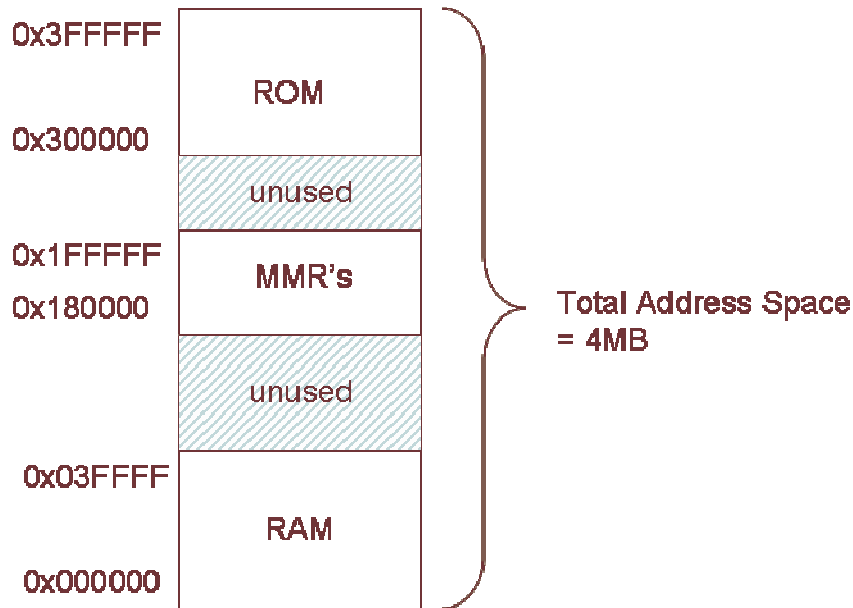
B. What does it mean when we say a signal is active-high or active-low? Are there any active-low signals in your memory system?



Active low means the signal has its specified action when in a logic low state. Active low signals are quite common in IC's (legacy from the bi-polar (TTL) days). There are two active low signals in this interface,  $\overline{WR}$  and  $\overline{RD}$ .

### 4. (10 points) Memory Map

A microprocessor generates 22-bit memory addresses. Draw a byte-wide memory map of its address space with boundaries specified in hexadecimal. A 1MB ROM is located in the memory address space ending at the highest possible address. A 512kB block of memory mapped I/O is located at base address 0x180000. A 256kB RAM is located at the lowest memory addresses. Show all of these devices in the memory map with their boundaries labeled in hexadecimal. How large is the total memory space?

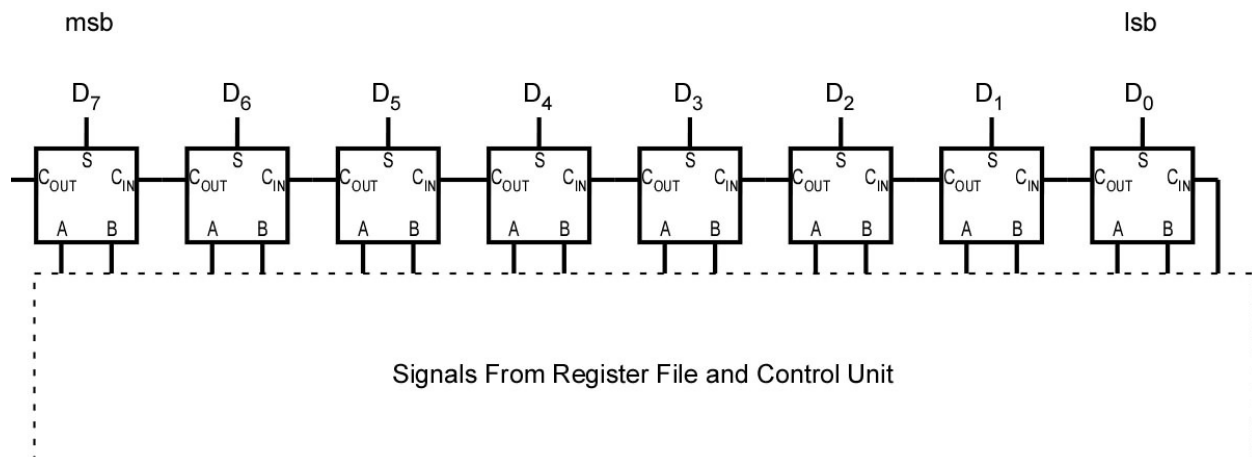


### 5. (15 points) Flags Register

Microprocessors typically have a flags register (also commonly called a condition codes register or processor status register) where certain information about recent operations is stored.

Common information includes whether there was an arithmetic overflow (for unsigned and signed operation), if the result was negative, and if the result was zero. (In the ARM7, these correspond to the CPSR register's C (unsigned overflow), V (2's-complement overflow), N (negative), and Z (zero) bits.) Assume that a microprocessor's ALU has an 8-bit adder as shown below.

- Draw logic to show how the four ARM flag bits can be efficiently generated.
- Explain how arithmetic overflow is detected for unsigned and signed operations, with examples.

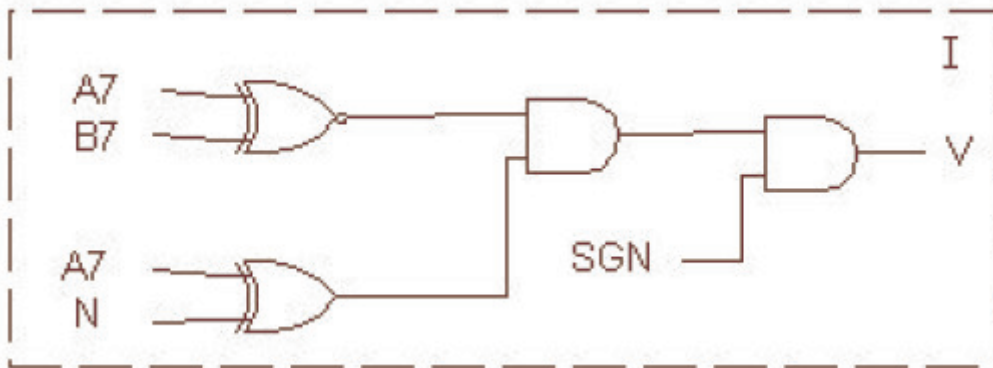
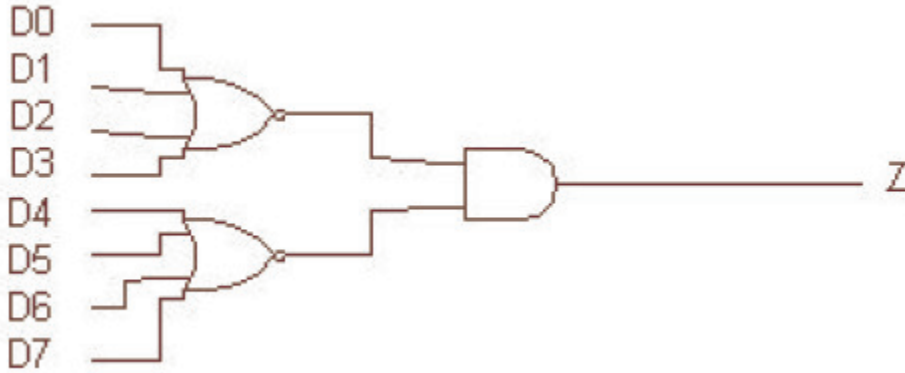


- The logic for the four flags is shown below. The Z flag is a NOR of all the bits. The N flag is the most significant bit of the result, since in a 2's-complement representation all negative numbers have a leading 1 bit. The C flag is just the carry out of the adder. The implementation of the V flag is shown in two variations. Box I is perhaps a more intuitive implementation, based on the idea that overflow can be detected when two values of the same sign have a sum with a different sign. If  $A_7$  and  $B_7$  are opposite in sign, then the

XNOR gate will have an output of 0, and V=0 regardless of the condition of the XOR gate or the signal SGN (which is 1 if the operation is signed arithmetic, 0 if unsigned). So, for A7 and B7 of opposite sign, no overflow can occur. If A7 and B7 are of the same sign, then the XNOR gate will have an output of 1 and the V flag will be 1 only if N (sign bit) is opposite of the state of the operand signs and the SGN bit is set, indicating that the arithmetic is signed. Box II implements a more efficient way to generate the signed overflow logic. For a detailed explanation, see pp. 220 – 221 of Logic and Computer

B. Unsigned arithmetic overflow is detected by a carry out of the adder. However, a carry out of the adder does not indicate that a signed overflow occurred, so we need to use the V flag logic instead. In the examples below, overflow is indicated in **red**.

	unsigned	signed		unsigned	signed
0xFF	255	-1	0x7f	127	-127
+0x01	+ 1	+ 1	+0x7f	+127	+ -127
0x00	<b>256</b>	0	0xFE	254	<b>-254</b>
C=1			C=0		
V=0			V=1		





**6. (10 points) Memory Endian-ness**

The value 0x3FC84 is to be stored in memory as a 32-bit value at address 0x0004 0128. Using small segments of a byte-wide memory map, show how it will be stored into memory in both a little-endian and big-endian system.

0x00	0x0004012B	0x84
0x03	0x0004012A	0xFC
0xFC	0x00040129	0x03
0x84	0x00040128	0x00
	Address	

Little Endian

Big Endian

**7. (15 points) Microprocessor Architectures**

A. Explain the fundamental difference between a Von Neumann and a Harvard architecture.

- i. List an one advantage and one disadvantage of each type
- ii. A PIC18F1320 is an example of what type (find the data sheet on-line)
- iii. Name a processor of the other type (opposite of what the PIC18F is)

Harvard	Von Neumann
+ Both data and instruction memories can be accessed simultaneously	- Data and instruction memory accesses must be interleaved
- Harder (or even impossible) to include literals or look-up tables in code section of memory	+ Constant data stored in code section of memory can be accessed in the same manner as instructions from the same area
- Two memory busses → uses more power	+ One memory bus → uses less power
+ Two memory busses → faster	- One memory bus → slower
Data and program memory widths can be exactly what is required.	

A PIC18F1320 is a Harvard architecture.

A processor of the other type (Von Neumann) is the ARM we study in this class

B. What are the relative advantages/disadvantages of RISC versus CISC architectures?

CISC	RISC
- Variable-size instructions → more complex decoding in hardware	+ Standard size of instructions → easier decoding in hardware

+ More functionality per instruction → fewer instructions per program	- Simpler instructions → more instructions per program
- Different instructions may execute in different lengths of time → harder to pipeline	+ Instructions take about the same time to execute → simpler pipelining
- Lower clock speeds	+ Higher clock speeds

### 8. (10 points) Quiz Question Development

Design one original quiz question operating at Bloom's Taxonomy level 3 for any material covered in Module 1. This must test one of the module objectives in a specific problem.

**Explicitly state which particular objective you are attempting to test.** Provide a complete and detailed solution to your question.

This question is typically graded on the following scale:

I love your question/answer and will use it in the future as a quiz question = 10

Decent question and **correct** answer = 8

Decent question but incorrect answer = 4 to 6 depending on how wrong your answer was

Obviously little or no thought put into question or answer = 0 to 2