

CS150, Fall, 2006, Midterm 2, Pister

NAME _____

SID _____

1) You are designing a 7 bit adder for a particular embedded system. Based on what you know of the requirements of the system, you decide to use a slightly different version of the normal 2s complement number system. You assign $1/8^{\text{th}}$ of the numbers to be non-negative, and $7/8^{\text{th}}$ of the numbers to be negative. You make a simple ripple carry adder.

A) In your number system, what is the binary representation of the numbers -1, 0, and 1?

B) What is the most positive and most negative number that you can represent (give your answer in both decimal and binary) ?

C) List all of the types of operand/result combinations that would indicate overflow.

D) Design the circuitry for NCVZ outputs (assuming the ripple carry adder is already done).
N=result is negative; C=carry out; V=overflow; Z=result is zero.

2) Design a triple-port register file with 16 8-bit registers. Port A should be read-only with an output enable, and ports B and C should be write only with ENABLEs. If both B and C inputs are trying to write to the same register, the B value should be written to the register and a C_WAIT line should be asserted if the B and C values are different. Do a top-down design starting with a high-level block diagram and working your way down to flip flops and gates.

Examples of what the register file can do in a single cycle:

- * read R2 onto A, write B data to R1

- * read R7 onto A, write B data to R0 and C data to R9

- * write B data to R3, write C data to R3. Since both the B port and the C port are trying to write the same register, this will cause B data to be written to R3, C_WAIT should be asserted if C data is not the same as B data.

3) Assume that you have a 16-bit ripple carry adder, and an SRAM organized as 1024x16 with CS and WE control inputs (similar to the one in CLD2). Draw a block diagram of a system to add memory location 0 to memory location 1 and store the result in memory location 3, then add locations 4 and 5 and store in location 7, and so on throughout the memory until $M(255) = M(252) + M(253)$. In general: $M(4*I + 3) = M(4*i) + M(4*i+1)$ for $I = 0$ to 255.

The system should start this operation when a RESET line is asserted, and it should stop when the last result is written to $M(255)$. Note that some memory locations are neither read nor written (e.g. locations 2, 7, 254).

- a) Clearly show the components and wiring of the datapath. You may use the adder, SRAM, registers, counters, MUXes, and logic gates.
- b) List the control lines in the datapath.
- c) Show a state diagram of the controller for the system.
- d) How would you modify your design if you needed to store the absolute value of the sum instead of the sum (assume 2's complement) ?

4) In a carry lookahead adder, each of the individual bits generates a P and G signal @1 gate delay. Design a level 2 carry lookahead circuit to make a 64 bit adder. Your design should consist of 9 identical carry lookahead blocks, each with 8 P and G inputs. Write down the equations for the outputs of the carry lookahead block (you don't need to write all of the equations if there is regular structure). Do NOT show the internals of all the blocks, or try to draw all 64 bits, or all the wiring. Just show enough of each so that the design is clear. **Clearly indicate when (@1, @2, etc.) the group P and G signals are generated, and when C8, C16, C56, and C64 are available.**

5) Short answer questions:

When do you send the first video request to the videoram in order to get data in time to send the first active video data in a given line?

Why is a random wait after a negative CCA check necessary?

What is the compression ratio from YCrCb sub-sampling in the ITU-601 standard?

What is YUV coloring and why is it used in televisions?