

## CSCE 230, Fall 2009

### Homework 4 (Appendices C & D, Chapter 3)

Due: Wednesday, 10/7/2009 before class (by Web Handin or hard copy)

Problem 1. [25%] Consider the input/output view of a 32Kx8 integrated-circuit (IC) chip shown in Figure C.9.1 of the textbook. The memory chips are designed so that it is easy to build larger memories using multiple instances of the same device, possibly, with addition of a small amount of extra logic. For this problem you are asked to use this 32Kx8 SRAM memory chip to build two larger memory devices as follows:

1. A 64Kx8 SRAM memory
2. A 32Kx16 memory

For full credit you should clearly show and document your design in each case.

Problem 2 [15%]. Cache memories are said to be *associative* because they store a part of the address of a word (and a few additional control bits) along with the data stored at that location. Assume the word size of a computer is 32 bits and the cache overhead is 20 additional bits per word, for a total of 52 bits. Further assume the cache is capable of storing 64K words of data.

- (a) You are given 32Kx8 SRAM chips to build the above cache memory. This means that you can only build larger memories  $M \times N$  such that  $M$  is a multiple of 32K and  $N$  is a multiple of 8.
- How many chips will be needed to build the cache? Justify your answer.
  - Because of the size constraint mentioned above, some bits in the cache will go unused. What is the percentage of wasted bits? Justify your answer.
- (b) Repeat part (a) given the size of the SRAM chips is 8Kx32.

Problem 3 [10%]. Draw an input/output view of 64Kx8 and 8Kx32 SRAM chips, similar to Figure C.9.1 of the textbook.

Problem 4 [20%]. Design a gate-level circuit for a modulo-8 counter that is capable of counting up or down when enabled. The counter has three binary control signals, called *Reset*, *Enable*, and *Up*. The Reset signal initializes the counter to the 0 state, independent of the value of other inputs. Otherwise, when the Enable signal is 0, the counter maintains its current value indefinitely; when Enable is 1, it counts up or down at the next rising edge of the clock depending on if Up is 1 or 0 respectively. Follow the systematic approach to designing finite state machines as discussed in the class, noting however, that because of the above behavioral description, the state assignment step is not necessary here. Show the truth tables of the next-state and output variables and convert them into a logic schematic using as few gates as possible.

Problem 5 [15%] Figure 3.5 in the text shows the flow chart of the “first multiplication algorithm” that most of you would have implemented in Lab 5. Figure 3.6 shows a refined version of the data

path that optimizes the needed hardware (compare it with Figure 3.4 which is the basis for the flow chart in Figure 3.5).

Assume the word size to hold unsigned binary operands is 4 bits (instead of 32 in Figure 3.6). Show the steps of multiplication algorithm working on the data path in Figure 3.6 for the following decimal numbers:

Multiplicand: 11

Multiplier: 13

Your answer should show the initial states of the Multiplicand and Product registers and the state of the Product register after each step of multiplication. Note that the double-length Product register, in this case, will be 8 bits wide.

Problem 6 [15%] Figures 3.9 and 3.10 show the data path and the hardware algorithm using it to perform the division operation. Read this section and the example to understand how the division is implemented in hardware.

(a) Do Exercise 3.7, part (a), from the text to demonstrate your correct understanding of the algorithm.

(b) Show an input/output view of the control unit for the division algorithm based on Figures 3.9 and 3.10.

(c) Derive a state-table description of the control unit, showing clearly the control operations performed in each state.