Homework Exercises Week 3: Binary numbers and Sequential Logic

IPC006 Processoren

Hand in before Monday, November 26 2018, before 20:00 via Brightspace.

You *must* do the homework exercises in pairs.

Discuss the homework exercises of Week 2 in the exercise class.

Also, discuss the following exercises in the exercise class:

1. Perform the following calculations in 8-bit binary numbers (use 2's complement). Determine for each calculation the flags (C,Z,O,N) (carry, zero, overflow, negative). The overflow flag is set to `true`, if and only if, two positive numbers are added and the result is too big to store as an 8-bit number (making the result being interpreted as a negative number), *or* when two negative numbers are added and the result is interpreted as a positive number.
   (a) $54 + 78$
   (b) $81 - 37$
2. Consider a hypothetical machine that can be in one of the two states Ready or Active. The machine has two sensors A and B. When the machine receives a signal from sensor A, it will transition from Ready to Active, and remain in that state until it receives a signal from sensor B. It will then transition back to Ready. The machine is controlled using edge-triggered synchronous logic, so all transitions happen on the rising edge of the clock.
   (a) The behaviour of the machine can be modelled by a *Finite State Automaton*. A *state transition function* determines the behavior for such an automaton. Intuitively, for each state and input, a successor state is defined. Specifically, the automaton for the machine described above has a state transition function that specifies, for each state Q and value of the inputs A and B, what the next state Q' will be.
   Give the *truth table* of the transition function.
   (b) Derive a *minimal Boolean expression* for the transition function, and implement this as a (combinational) circuit with inputs A, B, and Q, and output Q'. What is the meaning of the terms in this expression regarding the states and the inputs?
   (c) Design a control circuit for the full machine, with inputs Clk, A, and B, and outputs Ready and Active. The state of the machine should be stored in a D flip-flop. For the purpose of this exercise, you may ignore the initialization of the circuit (i.e. you may assume that the circuit automatically starts out in a sensible state). The circuit

of part (b), defining the transition function, may be used as a subcomponent of this circuit. It has inputs A,B, and the current state, and as output the next state.

3. Suppose we are working in a 32-bit processor. How should the flags be set after computing $0 - (-2^{31})$?

Hand in the solutions to the following exercises via Brightspace:

4. Perform the following calculations in 8-bit binary numbers (use 2's complement). Determine for each calculation the flags (C,Z,O,N) (carry, zero, overflow, negative).
   (a) -100 OR 100
   (b) $7 - -13$

5. Sign extension is the operation, in computer arithmetic, of increasing the number of bits of a binary number, while preserving the number's sign (positive/negative) and value.
   (a) Eight bits are used to represent the value 1101 0110 (decimal -42), using 2's complement. Sign-extend this to 16 bits. (The number's sign and value need to be the same).
   (b) Twelve bits are used to represent the value 0101 0011 1001 (decimal 1337), using 2's complement. Sign-extend this to 16 bits.

6. The objective of this question is to design simple circuits for counting. Make sure to draw the circuits in a structured and understandable way. You are allowed (but are not obliged) to already use HADES to draw the circuit and to submit a screenshot.
   (a) Draw a circuit with two 1-bit inputs: a RESET signal and a CLK. It has one 3-bit output called COUNT. Using 3 flipflops, a 3-bit number is stored in regular binary format. This number is shown in COUNT. When RESET is low, at each clock tick, this number is incremented with 1 (modulo 8). When RESET is high, the number is reset to 0 on the rising edge of the clock.
   (b) Then, expand the circuit, so that it counts modulo 6: when 5 has been reached, the next value of the number is 0.