CEG 3155 Summary Sheet

Contents

Well this course is so unreasonable its ridiculous. Who knows what will get done. I have a lot of words I want to say to the administration of this course, but I will hold back. One thing is for certain, despite what they claim, uOttawa does not care about the health at all of their students. They expect students to dedicate their whole life to school, not sleep well, not eat well, and really sacrifice their whole health. Many people can deal with this, but some people cannot. This is not right for some people to feel like they are out of options and have to end everything because of the actions of the university.

Basically so far it is about asm design, and some fsm design. :-/

Also, this course has a lot of processes that in my opinion are quite hard to explain. They need to be done to understand them. So I don't think putting them here would really help me learn. But hey, I'll see.

1 Miscellaneous Topics

Really just a review of ITI1100 and CEG2136

2 VHDL

VHDL is a hardware description language. It is used to model hardware.

We can built at multiple levels such as the structural level, and the behaviour level. The behavioural level is very high level and uses `if/else`, and `while` loops. It is easy to program, but it is not very efficient when it compiles it to the actual gates to be used in hardware.

At the structural level, we only use basic `AND` gates, `OR` gates, `NOT` gates, `XOR` gates, etc. We connect the wires between gates. We can create **components** which have inputs and outputs, and are also made up of gates (behaviourally coded, or structurally coded).

VHDL is quite a powerful and complicated language, but is better explained through examples.

3 ASM Design

Algorithmic State Machine is a method to take a design and turn it into a circuit. We have two main parts, the datapath and the control path. The control path just simply takes in the inputs and states, and generates control signals for each state we are in. These signals are used by the datapath and the controlpath.

Some signals could be to shift a register, load a flip flop, enable a certain flip flop, etc.

The datapath is where the actual data goes. It goes in, comes out, and takes in all the control signals.

4 Arithmetic

5 FSM Design

Finite State Machine is a method to take a design with a finite number of states and turn it into a circuit.

6 Asynchronous Circuit Design

6.1 State Reduction

We can either use the standard state equivalence method, or use the new state fusion method.

6.2 State Assignment

When assigning states, we cannot have any states where going from one state to the other more than one bit changes. So we could not for example go from `000` to `110`. We could however go from `000` to `010` to `110`.

To easily do this, we create a transition diagram. Here we do not want to have any diagonals. If we have a diagonal we either have to eliminate it (by going along a different path) or add more states to account for it.

6.3 Hazards

Static hazards are when a signal briefly changes values and then goes back to the original value. Dynamic hazards are when a signal switches a few times before settling on a new value.

We want to avoid both types of hazards. We can do this by trying to include more implicants in the K maps, and by using circuits with only two levels of gates.

Figure 1: Hazard Types

7 Implementation Technologies

There were many different ways of implementing circuits before today's FPGAs.

8 Interfaces and Controllers

9 Processor Design

10 Digital Testing

11 Appendix