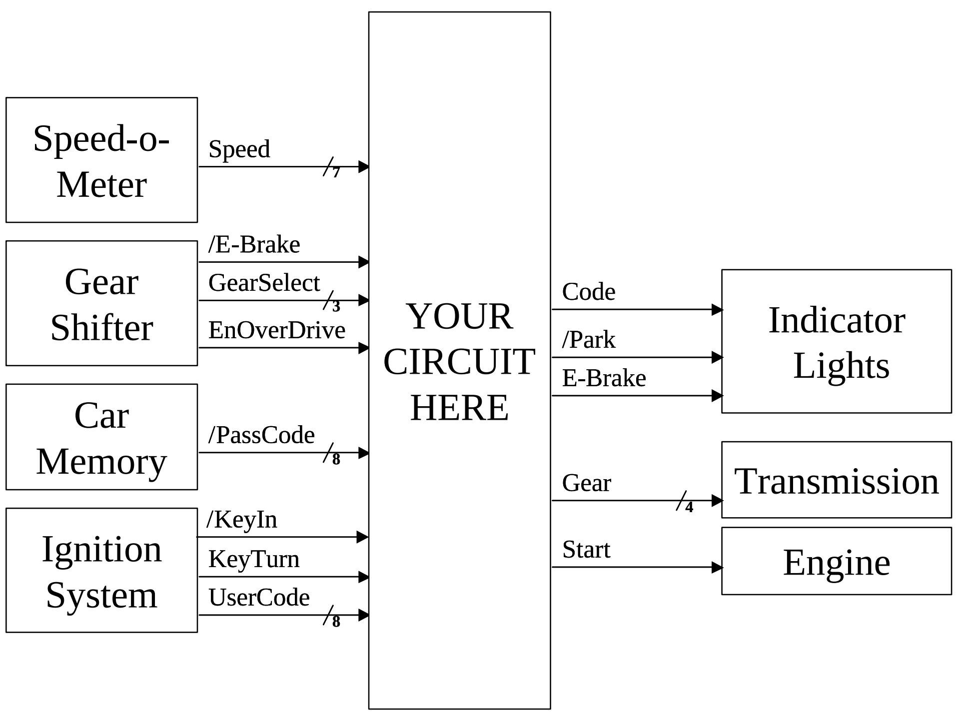
Fall 2003 – CSE 207 Digital Design Project #2

Background

Cool Star Engines has decided to move from just building engines for cars and will now be producing full automobiles. While the mechanical guys are figuring out how to bore out a passenger seat, you will be working with the computer and electrical engineers designing the car's electrical system. This car needs to be equipped with the finest power assisted components. Well, at least the finest ones that management could purchase in large quantity at really cheap prices. This means a few of the pieces have strange interfaces that just have to be designed around. The car being produced is fairly utilitarian. It has 4 tires, a steering wheel, an engine, a transmission, and a state-of-the-art super-duper security system. You must design the control circuit to make these pieces work together as described below.

Design Overview

The key switch has two sensors in it. The first one indicates that a key has been inserted into the ignition lock (/KeyIn). As the driver turns the key to start the engine, the KeyTurn signal is asserted. While the key is in the ignition, the user's security code most be verified against the code stored in the car's computer. The car has an 8-bit code (PassCode) stored in memory that needs to be compared against the code coming from the key (UserCode) when the key is turned. Because the memory is cheap, all of the bits come out inverted. To detect a match they must first be flipped and then compared. The car cannot be started if the key does not have the correct code. Additionally, the car cannot be started if the gear selection is in anything other than park or neutral. Start only needs to be asserted while the key is being turned. Once the user let's go, the key will only report that it is in and the engine will continue to run all on its own. The engine will stop on its own when the key is removed (really cheap parts ☺). The gear shifter has a 3-bit number that defines which gear the car is in. The values produced are 0=Unknown, 1=Park, 2=Reverse, 3=Neutral, 4=Drive, 5=SecondGear, 6=FirstGear. The over drive button allows the car to change into the 4th gear if the car is in Drive and the speed is greater than 60mph.



Part of the circuit you are building will include safety features. If you detect that the speed is greater than 25mph you will override the gear selection and tell the transmission to switch to 2nd gear. If the speed is greater than 50mph, again, the gear selection will be overridden and the transmission will be switched to 3rd gear. Normally the car should change to 2nd gear at 15mph and into 3rd gear at 40mph if it is in drive (but it won't switch if the person has selected 1st or 2nd gear only). The transmission will go to whatever gear it is told, so care must be taken to handle all invalid cases and put the transmission into a known state at all times. The possible states for the transmission are provided as the 4-bit word where F=Reverse, 1=FirstGear, 2=SecondGear, 3=FourthGear, 4=ThirdGear, E=Neutral, D=Park). Notice that 4th and 3rd are not in the order you expect. This is because on 3-speed models of this car, a 1-hot encoding scheme is used for the other bits when the most significant bit (MSB) is a '1'.

Lastly, some indicator lights can help the user figure out why the car will not start. These are the Code, Park, and E-Brake lights, which inform the user that they have an invalid key, need to put the car in park to start it, and should turn the e-brake off if it is on. Pay attention to lines that are active low (i.e. they are asserted when the value is a '0' – normally we work with active high indicating at the signal is asserted when the value is a '1').

Considerations

This circuit is purely combinational. This means that your circuit has no concept of time and cannot remember past events. Whatever the inputs are, you should be able to determine the outputs. You will set the PassCode using two hex keypads in LogicWorks. You will also set the UserCode from the ignition system using the same set up. Use hex displays to make it easier to read multi-bit numbers (i.e. the gear into the transmission system). This design is larger than the 1st project. While you are more restricted on the inputs and outputs of this design you are free to break your design it into any number of modules so that you can meet the requirements. In doing so, make sure you consider several options before you try to build this and document a few of the trade offs. You will probably find that there is more than one way to partition your circuit and will find that the design can be simpler or more complex based on different choices. If you have a question about the spec, please ask – and check the web page for updates as your colleagues catch problems with the requirements. The bottom of the page has the date that the document was last modified – make sure you are working from the most up-to-date spec.

Due Dates

September 24th, 2003: Initial Design – Pen and paper work only at this stage. I am only looking to see that you have started to break the big circuit into smaller circuits that have clearly defined inputs and outputs. Think about the approach you will take to build the circuit and each of the blocks. If a block looks to big, try to break it down more or use an ad-hoc design approach.

October 1st 2003: Functional Design – At this point, you should have a clear picture of all blocks and for the ones you are the least confident with, you should be working towards a schematic. Eliminate the "high risk" items first and then put the "easy" blocks together. You can work on the module sections of your report as you are doing this. Bring any materials you have to class so your TA can see your progress. *Circuits within blocks must be reduced if possible.* Show work for how you reduced them. If k-maps were used, these diagrams must be included. Reductions can either be in the main report or as an appendix that is referenced in the main part of the report.

October 8th, 2003: Project Due – The completed project along with report must be submitted as specified in the course information. Please be sure to follow the submission requirements. Remember, printed copies of your report are required along with the electronic version. A full formal report is expected, including a title page with your name, course number, section, and date. The body of the report must include a brief description of the problem you are solving (objective), a top-level design that your manager can read and understand, and a detailed description of each of the top-level blocks. For each detailed block, there should be a set of truth tables. These truth tables need to be converted to the equations actually implemented. A copy of the circuit that is implemented from these reduced equations should follow and be referenced so the reviewing panel can understand your design. Reports must be word processed and pieced together into a single final document. Schematics can be cut and pasted in the report. If you have questions on how to do this email me or see your TA. If time permits, you will be asked to demonstrate your circuit for your TA on the due date.