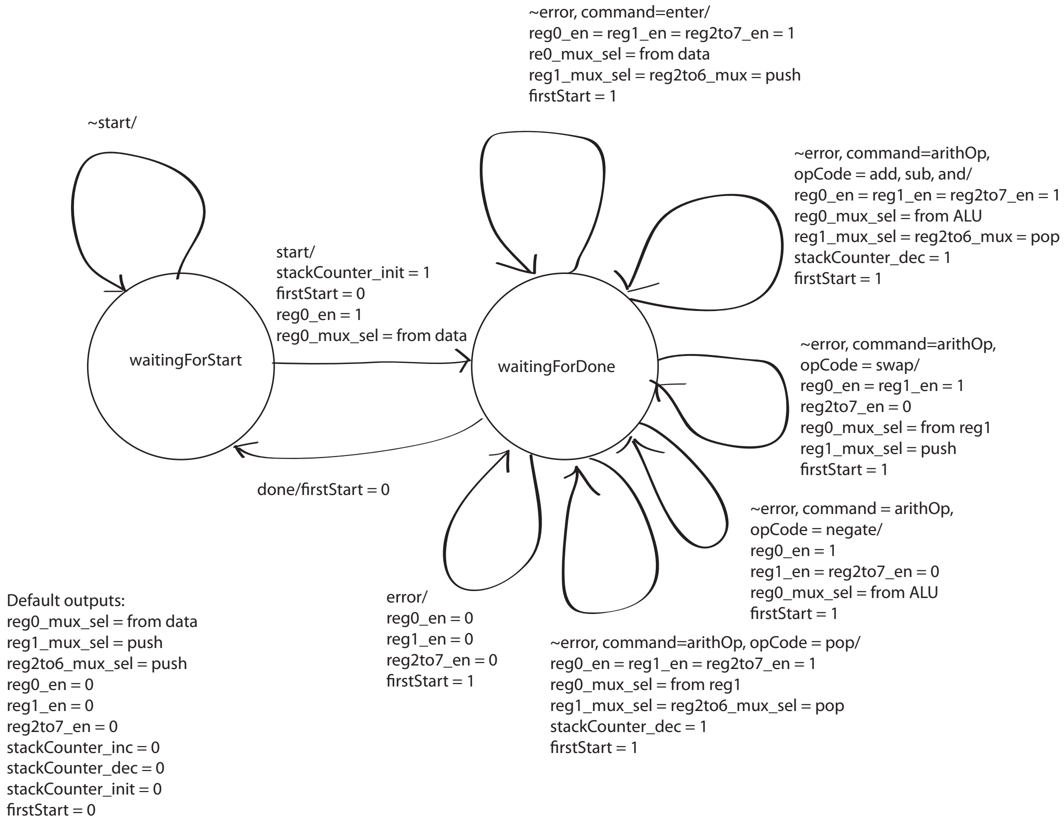
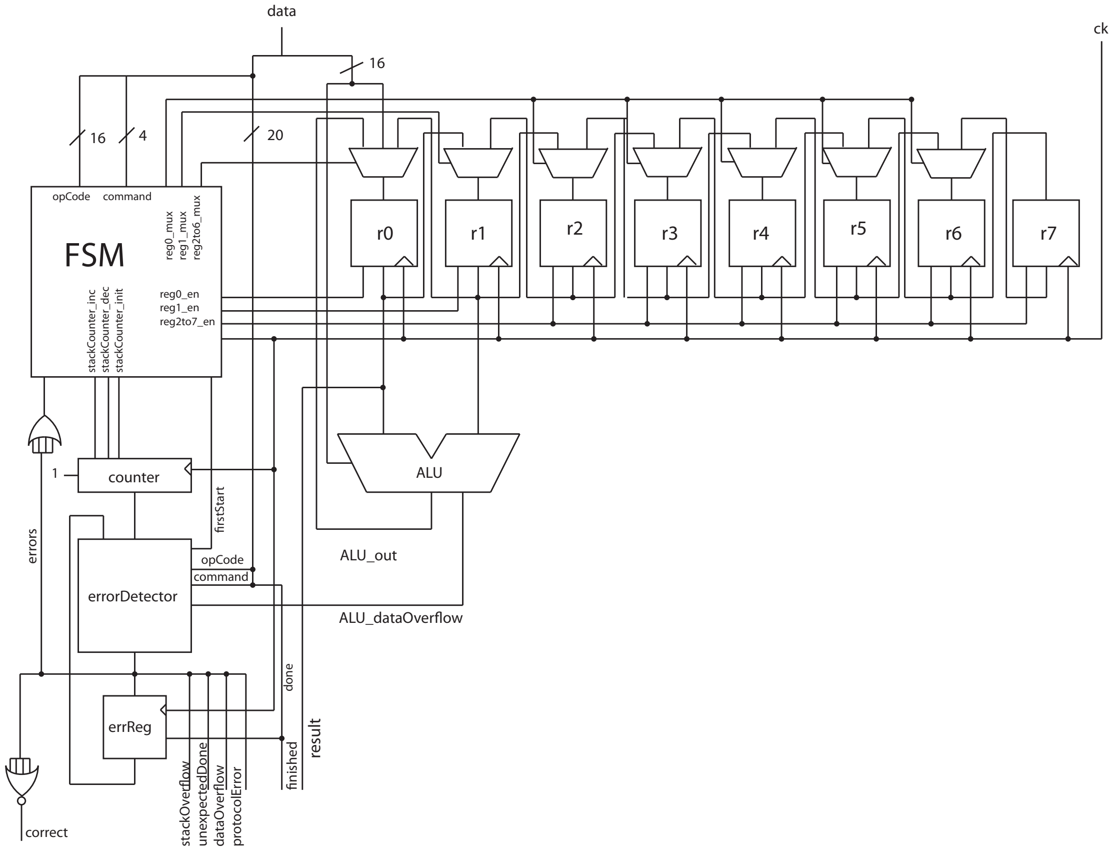
Paul Kennedy
18-341
Project 1
25 January 2011

State Transition Diagram



Note: for each transition, if a particular output is not specified, it is the default given in the list to the lower left. For example, when transitioning from waitingForStart to waitingForDone, reg2to7_mux is 0. Thus, all outputs are specified in all transitions and the FSM is therefore combinational.

As the names imply, the first state is simply waiting for the start command, while the latter state is waiting for the done command. Each loop infinitely until seeing their corresponding command. The waitingForDone state loops with or without errors (though once an error has been asserted, it can only be cleared by asserting done).



Above is the FSM and datapath for the calculator module. The FSM has 10 outputs: stackCounter_inc, stackCounter_dec, stackCounter_init, reg0_en, reg1_en, reg2to7_en, reg0_mux_sel, reg1_mux_sel, reg2to6_mux_sel, and firstStart. The stack consists of 8 registers where the first two registers are permanently hardwired into the inputs of the ALU. When new values are entered, each value in the stack is pushed down to the next register and the new value is loaded into r0. The first register can receive input from either the ALU output, the incoming data, or the r1 output. The rest of the registers (excluding r7) can take their input from either the previous or next register in the stack.

The errors are determined using the errorDetector module, which employs combinational logic on the inputs from various parts of the datapath. These errors are then immediately asserted on the outputs and stored in a register which itself feeds its output back into the errorDetector module.