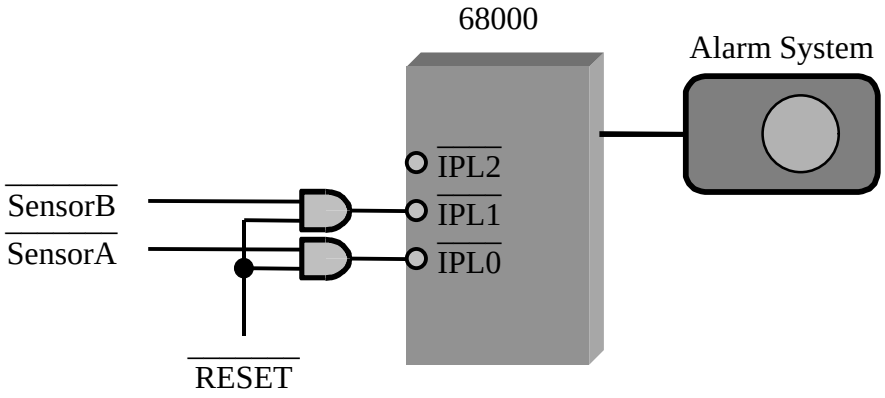
Dr. Bassem Al-Halabi, S&E362 July 28, 2000

Name:

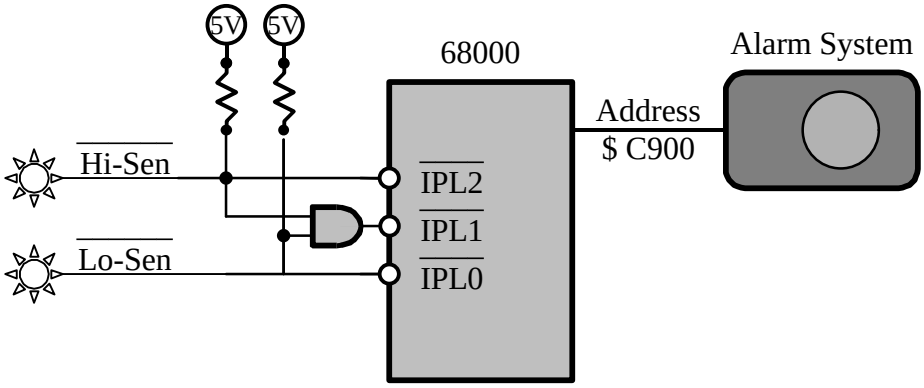
Sequence Order Using Semaphores

[10] **1**) Consider the following external interrupt circuit of a simple 68000 system with two sensors. The two sensors detect two different events: A and B. When event A occurs, an active-low pulse is generated by SensorA, and when event B occurs, an active-low pulse is generated by SensorB. Write a complete 68000 assembly program which endlessly monitors the sensors so that events A and B occur alternatively. As long as events A and B alternate, no action is taken. If at any time one event occurs for a second time before the other event, a special warning signal is generated by TRAP#8. This warning signal will be repeatedly generated every minute regardless of any subsequent A or B events. The system will stop calling TRAP#8 (stop warning) and return to its normal course when the RESET button is pressed. Assume that MINT_DLY subroutine and TRAP#8 are already available for you.



Water Level Monitoring Text/Audio Messages

[10] **1**) Consider the following external interrupt circuit of a simple 68000 system which is installed at a certain point of a river. When the water level is too high, sensor Hi-Sen asserts a zero. When the water level is too low, sensor Low-Sen asserts a zero. (You will never have both sensors activated at the same time.) Write a complete 68000 program, which endlessly monitors the sensors. When the water level is too high, a voice message "water_too_high__" is generated. When the water level is too low, a voice message "water_too_low__" is generated. TRAP#10 is already available for you, which reads a text message from memory pointed to by A0, converts it to voice and sends it to the Audio System.

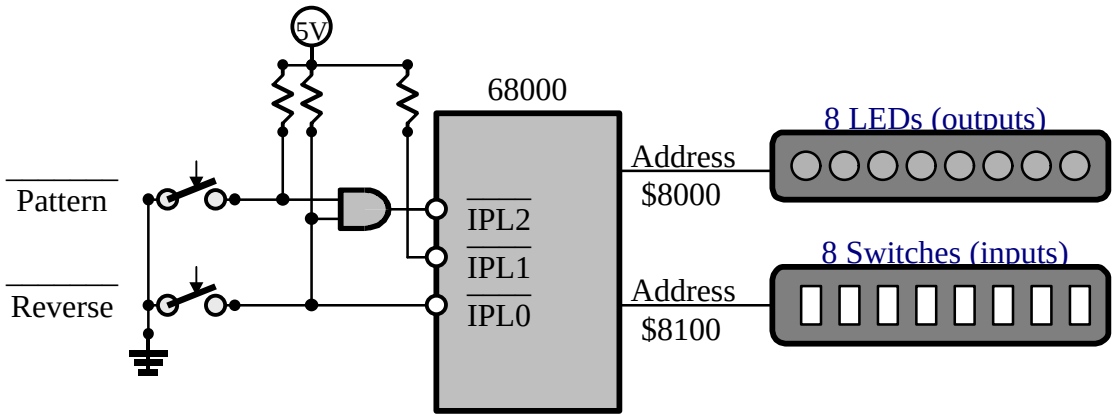


Dr. Bassem Al-Halabi, S&E362 July 28, 2000

Name:

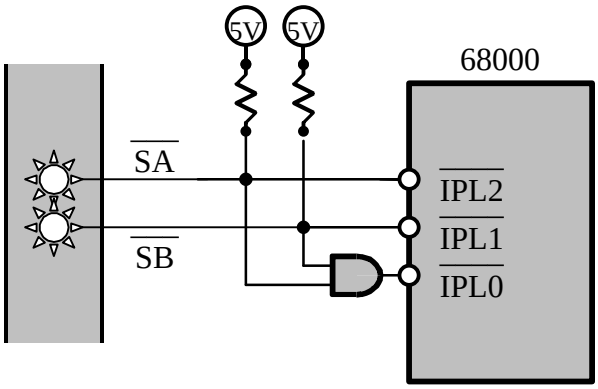
Pattern Read, Manipulate, and Display

[10] **1)** Consider the following 68000 system, which reads an 8-bit pattern from input switches, mounted on address $8100 and displays it on the 8-bit output LEDs mounted on address $8000. The system endlessly rotates the pattern on the LEDs. Use a DELAY subroutine to slow down the display time. When Pattern interrupt switch is depressed, the 68000 will execute an exception, which reads a new pattern from the switches. When the Reverse interrupt switch is depressed, the 68000 will execute another exception, which reverses (toggles) the direction of rotation (left/right). Write a complete 68000 program (from ORG to END) including the two exceptions. Do not write the DELAY subroutine, just use it.



Traffic Flow Control

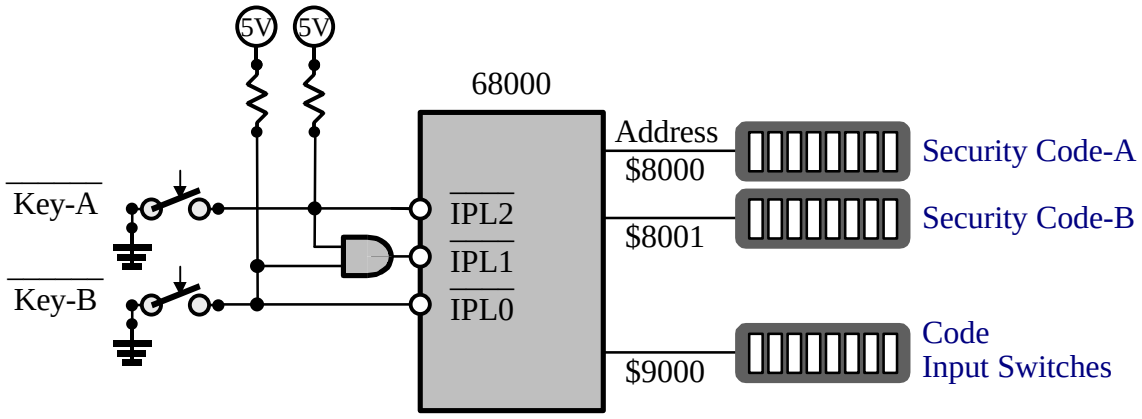
[10] **1)** Consider the following 68000 system, which controls the traffic flow on a one-lane road. If a car is moving south, SA detects an active-low pulse, which causes an interrupt (INTSA), and then sensor SB detects an active-low pulse, which causes another interrupt (INTSB). Now, if the moving vehicle is long, such as a bus, it will hit SA first (INTSA) and then SB but before releasing SA, a situation which is designed to cause yet another interrupt (INTBOTH). Similar analogy applies for vehicles moving in the north direction. Write an assembly language program, which keeps four (4) counts, the number of cars moving south, cars moving north, buses moving south, and buses moving north.



Name:

Security Gate with Keypad

[10] **1)** Consider the following 68000 system, which controls a tight security gate. The system is equipped with 2 8-bit fixed security codes, Code-A and -B, which are mounted at addresses $8000 and $8001 and set up by a system operator. To open the gate, two persons who have memorized these codes must be present at the gate. The first person enters his code on the Code Input Switches (mounted at address $9000) and turns his key, Key-A, which invokes an interrupt that reads his code and compares it to the first fixed code, Code-A. If there is a match, the program sets a flag for possible admission. Then similarly, the second person enters his code on the Switches and turns his key, Key-B, which invokes another interrupt that reads his code and compares it to the second fixed code, Code-B. If there is a match again, the program opens the gate by calling a special trap, TRAP#10, before returning from this second interrupt. Otherwise, (when either Key-A or Key-B, or both failed to get a match), the second interrupt will call TRAP#11 instead for alarm signals. TRAP#10 and TRAP#11 are already available.



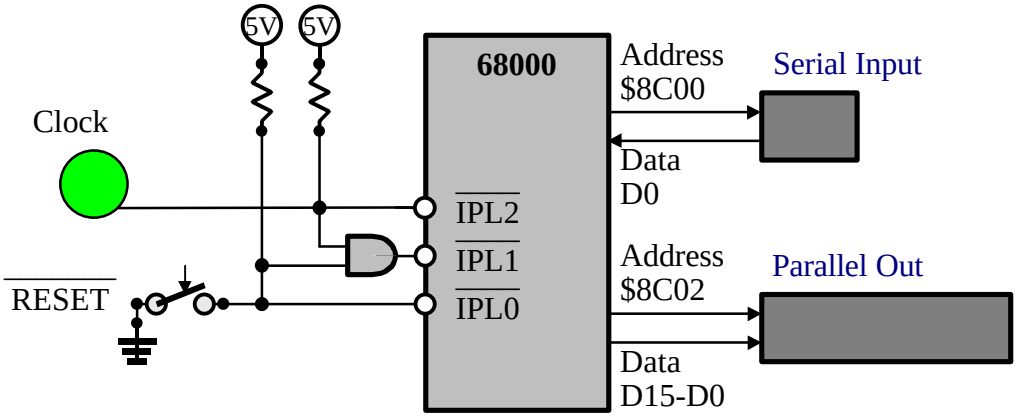
Hint:

Use counting semaphores

Name:

Serial to parallel Data Conversion

[10] **1)** Consider the following 68000 system, which converts serial input (Address $8C00) to parallel output (Address $8C02). The serial data is read one bit at a time from bit D0 from the data bus. This read process is synchronized by an interrupt generated by the Clock. After 16 of such interrupts a whole word is read in and sent out as a parallel output. Write a complete assembly language program that performs this conversion endlessly. The RESET button resets the conversion, which means the next coming bit on the Serial Input will be the LSB of the new 16-bit Parallel Output.



Hint:

Read one bit at a time and shift it into another register to build the whole word