Interrupt module notes:  
  
  
> although the control unit is to be much larger, I only use one chip as to symbolize the whole unit.  
All eeproms will receive the same input signal, so only one is needed to symbolize the cu  
  
  
  
Description:  
  
Typically the control unit starts off a new instruction by performing the fetch cycle and then said instructions microinstructions. However, if the interrupt request line is active and the current operating state allows for interrupts, the control unit will instead perform an interrupt instruction. Because an interrupt instruction must be able to occur during any form of instruction, every instruction must have interrupt code programmed into the cu.  
  
To prevent an interrupt from only partially activating, i.e. request occurs during micro instruction 011,  
there must be some sort of initial activation which acts as a prerequisite for an interrupt instruction.   
  
Part of an interrupt instructions first micro is to change the operating state into “interrupting”.   
If the interrupt occurs after it is able to enable this state, then the current operation will continue.  
  
To rephrase, the only operating state that an interrupt is able to occur during is the “interrupting” state.  
The only chance the operating state gets to change into the interrupting state, is when the current operating state is interruptible, the first micro is occurring, and an interrupt request is present.  
  
  
  
  
  
  
hierarchy chip:  
  
  
Interrupt requests from every module are sent to an eeprom’s inputs. From here, the eeprom will make a decision on which interrupt has the highest priority. (i.e a shutdown interrupt would have a higher priority than keyboard interrupt). After deciding on an appropriate signal, the eeprom outputs a binary value representing the selected interrupt.  
  
   
  
  
  
This value is fed into an address generator, which holds pointers to software based interrupt resolution.  
  
during an interrupt, this address is sent to the instruction module, via the cu.

The resolution lines lead to the module in question who prompted an interrupt. Upon resolving an interrupt, the CU will set the respective resolution line to one, which resets the interrupt sr latch