

Exceptions

Spim Input

- Spim allows you to read from the keyboard
- Receiver control bits tell the computer whether anything has entered the keyboard buffer
- 0 - no interrupt (like \n)
- 1 - triggers an interrupt (like endl)

Each data byte we need to read/write is mapped to an area of memory, and then we need to loop until the system is "ready with the data" by checking the "ready bit" periodically.

This is inefficient and is unsustainable, since the processor has to do many things, and is potentially running other programs

Interrupts

With external interrupt, if an event happens that must be processed the following will happen

1. The address of the instruction that is about to be executed is saved into a special register called EPC
2. PC is set to be 0x800000180, the starting address of the interrupt handler which takes the processor to the interrupt handler
3. The last instruction of the interrupt should be "eret" (similar to jr) which sets the value of the PC to the value stored in EPC (similar to ra)

Is it okay to use \$t0 in the interrupt?

- For an interrupt the user program is running and gets interrupted. The user program does not know about the interruption
- So if you changed 4t- inside an interrupt, after it returns the user program will not be aware, thus using the wrong value of \$t0.