Interrupts…

Devices (and especially) are slow compared to processors

When you interact with devices

- Polling (reading the status register repeatedly) is a waste of time
- Better to do other work if you have other work
- But need to be responsive to devices

Interrupts are like doorbells

When a device raises an interrupt, processor immediately pays attention

In the state machine for LC-3, for example, the first fetch stage goes off to handle an incoming interrupt

Handling an interrupt requires

(1) Saving all state of the processor (the code executing doesn't expect the interrupt), including condition codes

(2) Execute an interrupt handler (a subroutine). In LC-3, there's an interrupt vector table x0100 to x01FF, save return address to stack (do not overwrite R7), then PC <- M[0x100 + ZEXT(int-vec-8)]

(3) Subroutine (interrupt handler) ends in RTI, which restores state including condition codes