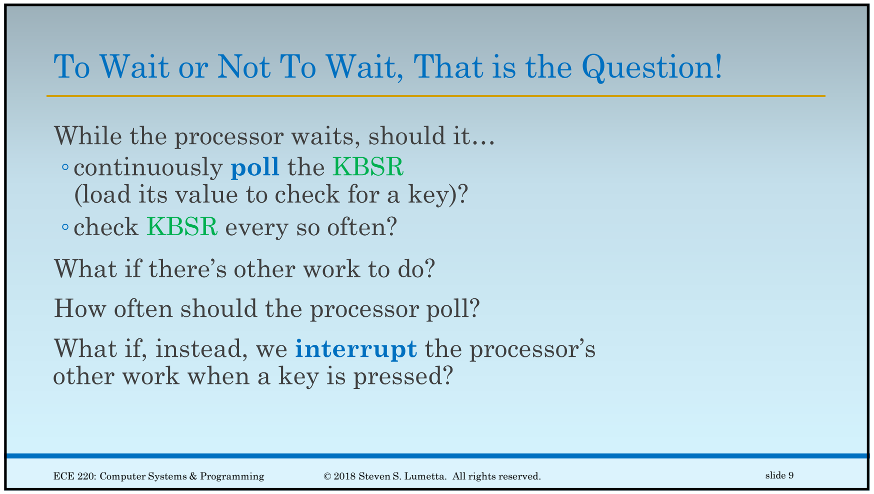


To Wait or Not To Wait, That is the Question!

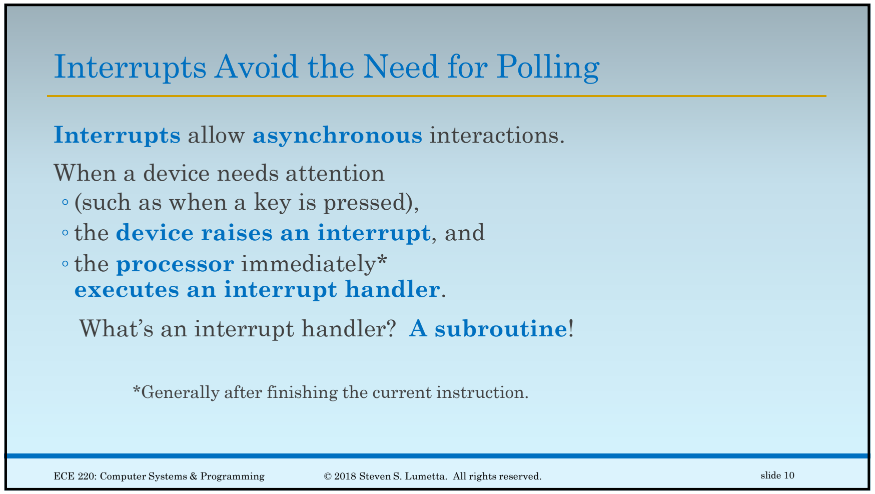
While the processor waits, should it…

- continuously **poll** the KBSR (load its value to check for a key)?
- check KBSR every so often?

What if there's other work to do?

How often should the processor poll?

What if, instead, we **interrupt** the processor's other work when a key is pressed?



Interrupts Avoid the Need for Polling

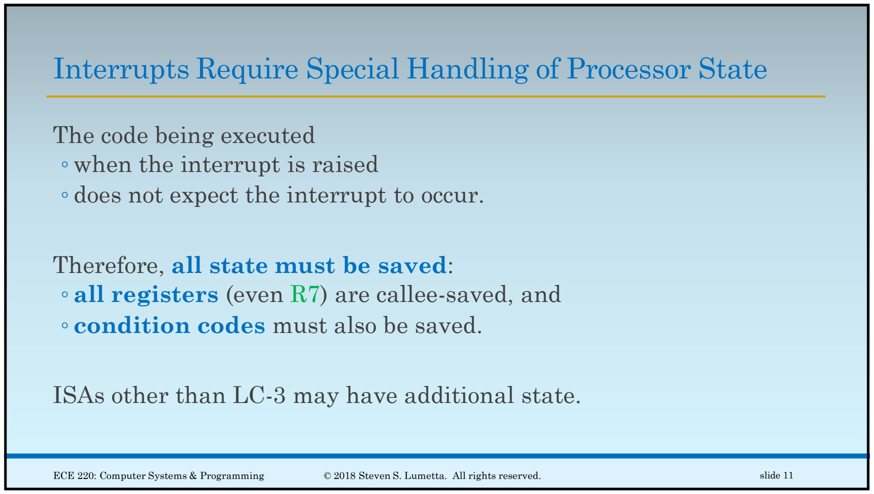
Interrupts allow **asynchronous** interactions.

When a device needs attention

- (such as when a key is pressed),
- the **device raises an interrupt**, and
- the **processor** immediately* **executes an interrupt handler**.

What's an interrupt handler? **A subroutine**!

*Generally after finishing the current instruction.



Interrupts Require Special Handling of Processor State

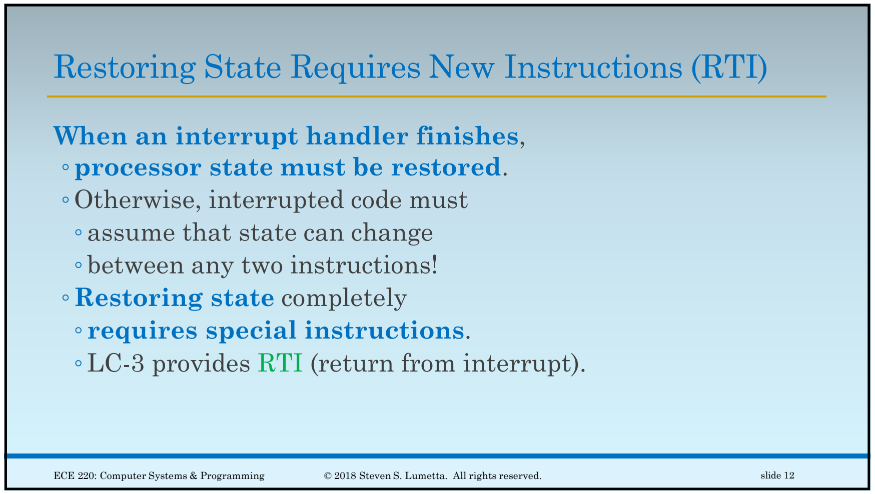
The code being executed

- when the interrupt is raised
- does not expect the interrupt to occur.

Therefore, **all state must be saved**:

- **all registers** (even R7) are callee-saved, and
- **condition codes** must also be saved.

ISAs other than LC-3 may have additional state.



Restoring State Requires New Instructions (RTI)

When an interrupt handler finishes,

- **processor state must be restored**.
- Otherwise, interrupted code must
  - assume that state can change
  - between any two instructions!
- **Restoring state** completely
  - **requires special instructions**.
  - LC-3 provides RTI (return from interrupt).