

Each Register is Either Caller- or Callee-Saved

(Remember: code **A** calls subroutine **B**.)

If a register might be changed by **B**,

- code **A** is responsible for
- copying the bits elsewhere before calling **B**.
- Such a register is **caller-saved**.

If subroutine **B** guarantees

- that a register does not change,
- that register is **callee-saved**.
- Note: **B can still use the register**, but must save and restore the original contents to do so.

R7 and Output Registers are Always Caller-Saved

In LC-3 subroutines,

- **R7 is always caller-saved.**
- JSR(R) changes it,
- so the subroutine cannot preserve its value.

Any output register

- **is also caller-saved,**
- since the subroutine changes the register's value
- as part of completing its task.

Also Document Any Side Effects

What are side effects?

- Any change to memory (other than to memory used exclusively by the subroutine)
- Any I/O performed by the subroutine (except for I/O specified as part of the subroutine's behavior).

Always document any side effects clearly.

Subroutine Calling Interface Includes Four Parts

Together, specification of

- subroutine inputs,
- subroutine outputs,
- ownership of other registers, and
- side effects

form the **call interface** (or calling convention*) for the subroutine.

*I find it strange to have a 'convention' for one subroutine, but you may hear it used.