
Computer Architecture III

AGENDA

- Guided Project Pt. II
- CPU Stack
- *I will have an AMA on Thursday 6-7PM PST*

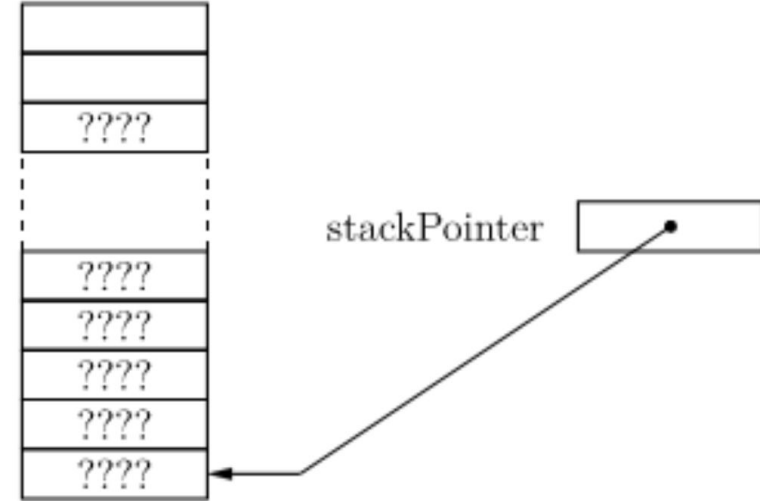
GUIDED PROJECT PT. II

- Implement the *load()* function to load .ls8 files
- Implement the *multiply* operation

CPU Stack

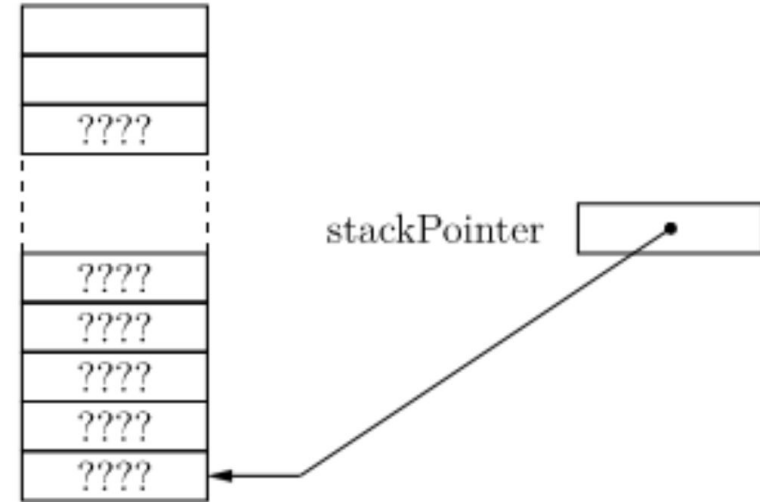
CPU STACK

- Used by the CPU to save state and store data
- **stack pointer (SP)** - a register (R7) that stores the address of the topmost element of the stack
- Two main operations:
 - **push** - decrease the stack pointer and write the element at the top of the stack
 - **pop** - read the element from the top of the stack and increase the stack pointer



CPU STACK

- What happens if you push too many times?
 - Stack overflow occurs
- What happens if you pop too many times?
 - Stack underflow occurs
- How can you detect if the stack is empty?
 - Check the address



CPU Stack Demo

GUIDED PROJECT PT. III

- Implement the CPU stack to your Is8 and be able to run *stack.ls8*

GODBOLT

- See the compilation output for a programming language!
- [Link](#)