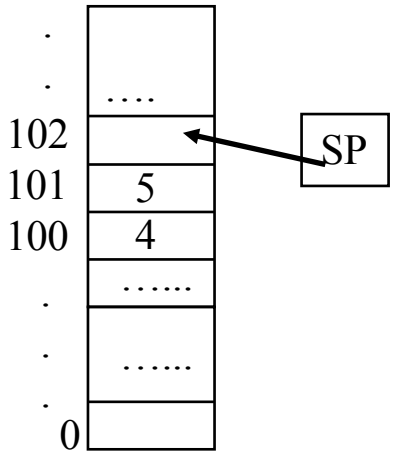
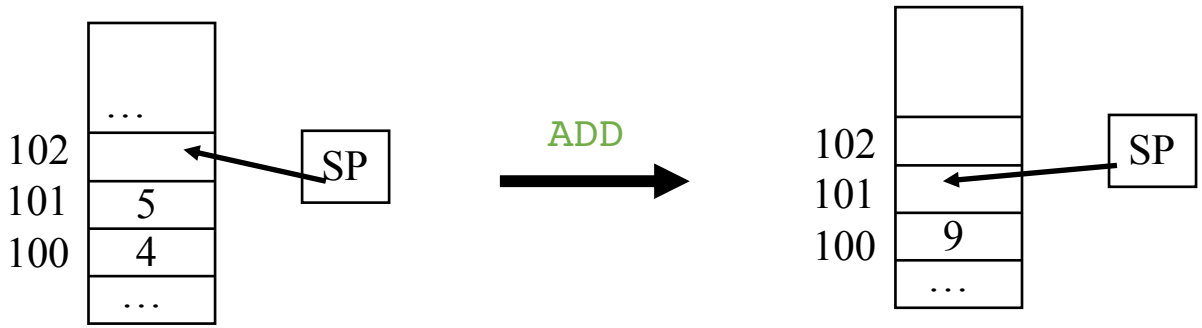
SaM Is A Stack Machine

- All data is stored in the stack or the heap.
  - No data registers. Three control registers (PC, SP, FBR).
- Stack also contains addresses.
- Stack pointer (SP) points to the first free location in the stack.
- Stack grows upwards from location 0.
- Atomic types and addresses take one stack location.
  - Strings and objects are stored in the heap.



Structure of Operations

- General invariant
  - Pop necessary number of operands from top of stack (TOS).
  - Push result value(s) to TOS.



Pushing Immediate Values on Stack

- `PUSHIMM c`
  - "push immediate": value to be pushed is encoded as part of instruction.
  - Pushes the value *c* to TOS.

ALU Commands

- `ADD`, `SUB`, …
- `DUP`: duplicate TOS.
- `ISPOS` (“is positive”):
  - Pop stack; let popped value be *Vt*.
  - If *Vt* is positive, push `true` (1); otherwise push `false` (0).
- `ISNEG` (“is negative”): Same, but tests for negative value on TOS.
- `ISNIL` (“is null”): Same, but tests for zero value on TOS.
- `CMP`:
  - Pop two values *Vt* and *Vb* from stack.
  - If (*Vb* < *Vt*), push 1.
  - If (*Vb* = *Vt*), push 0.
  - If (*Vb* > *Vt*), push -1.

Load/Store Commands

- SaM ALU commands operate on values at or near TOS.
- To operate on values from the interior of the stack, we need to copy them from an interior location to TOS, or write them from TOS to an interior location.

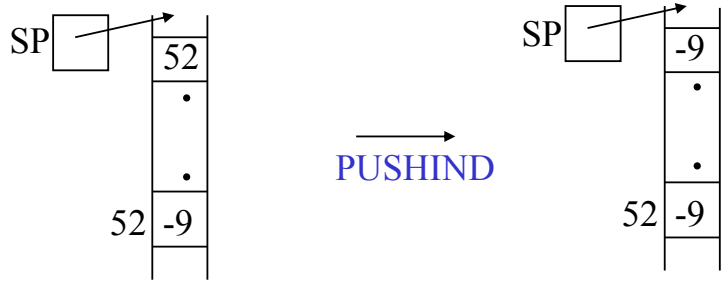
Load/Store Commands

- SaM ALU commands operate on values at or near TOS.
- To operate on values from the interior of the stack, we need to copy them from an interior location to TOS, or write them from TOS to an interior location.

- Two ways of specifying the address of an interior stack location
  - Indirect mode: The address is specified as a data argument on TOS.
  - Offset mode: The address is specified as part of the command and is interpreted as an offset from the FBR.
- Corresponding instructions:
  - `PUSHIND`, `STOREIND`.
  - `PUSHOFF`, `STOREOFF`.

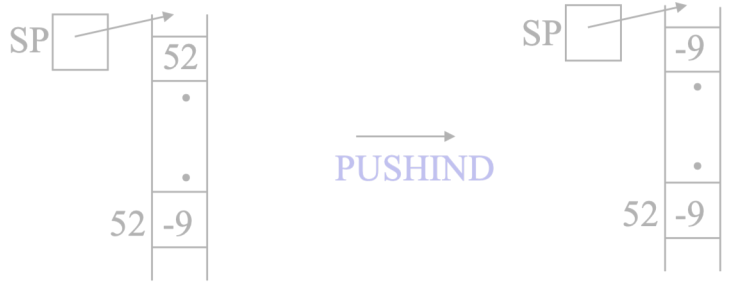
Load/Store Commands (Indirect Mode)

- `PUSHIND`
  - TOS contains an address.
  - Pop that address.
  - Read contents of that address.
  - Push contents on stack.

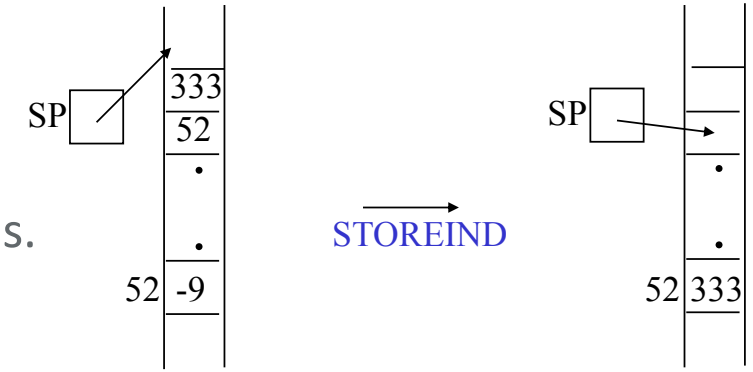


Load/Store Commands (Indirect Mode)

- PUSHIND
  - TOS contains an address.
  - Pop that address.
  - Read contents of that address.
  - Push contents on stack.

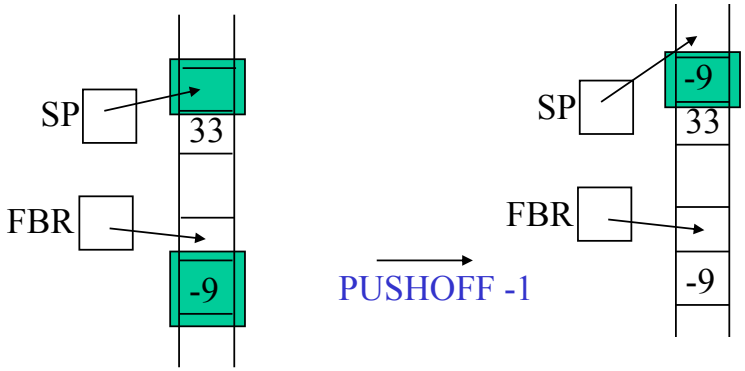


- STOREIND
  - TOS contains a value v.
  - Element below TOS contains address s.
  - Pop both.
  - Stack[s] ← v.



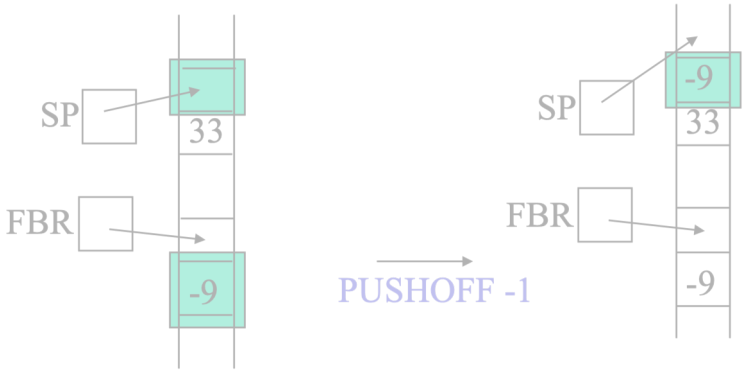
Load/Store Commands (Offset Mode)

- `PUSHOFF` *n*
  - v ← Stack[FBR + *n*].
  - Push v to TOS.



Load/Store Commands (Offset Mode)

- `PUSHOFF` *n*
  - v ← Stack[FBR + *n*].
  - Push v to TOS.



- `STOREOFF` *n*
  - TOS contains a value v.
  - Pop it.
  - Stack[FBR + *n*] ← v.