

Stack Control in Motorola 680X0

Definition of a Stack

A **stack** is a sequence of storage locations that are accessible from only one end, referred to as the **top of the stack**.

Basic Stack Operations

- **Push:** Inserts an operand onto the top of the stack.
- **Pop:** Removes the operand from the top of the stack.

Both operations modify the **stack pointer (SP)**, which is an address register that always points to the current top of the stack.

Stack Behavior

- In a push operation, the stack pointer is decreased before the operand is stored.
- In a pop operation, the operand is read before the stack pointer is increased.
- The amount of change in the stack pointer depends on the size of the operand (e.g., 4 bytes for a long word).

Stack Control in Motorola 680X0

The Motorola 680X0 does not include specialized hardware support for stacks. Instead, it leverages its flexible addressing modes.

Any general-purpose address register can be used as a stack pointer. For instance, we can designate A2 as the stack pointer, and assume the stack grows toward lower memory addresses.

Push Operation Example

To push the contents of data register D6 into the stack:

MOVE.L D6, -(A2)

This instruction is equivalent to:

$$A2 := A2 - 4 \quad \text{and} \quad M[A2] := D6$$

Pop Operation Example

To pop the top of the stack into D6:

MOVE.L (A2)+, D6

This instruction is equivalent to:

$$D6 := M[A2] \quad \text{and} \quad A2 := A2 + 4$$

Stack Operation Illustration

Prior to executing `MOVE.L D6, -(A2)`:

- $A2 = 0x1000$

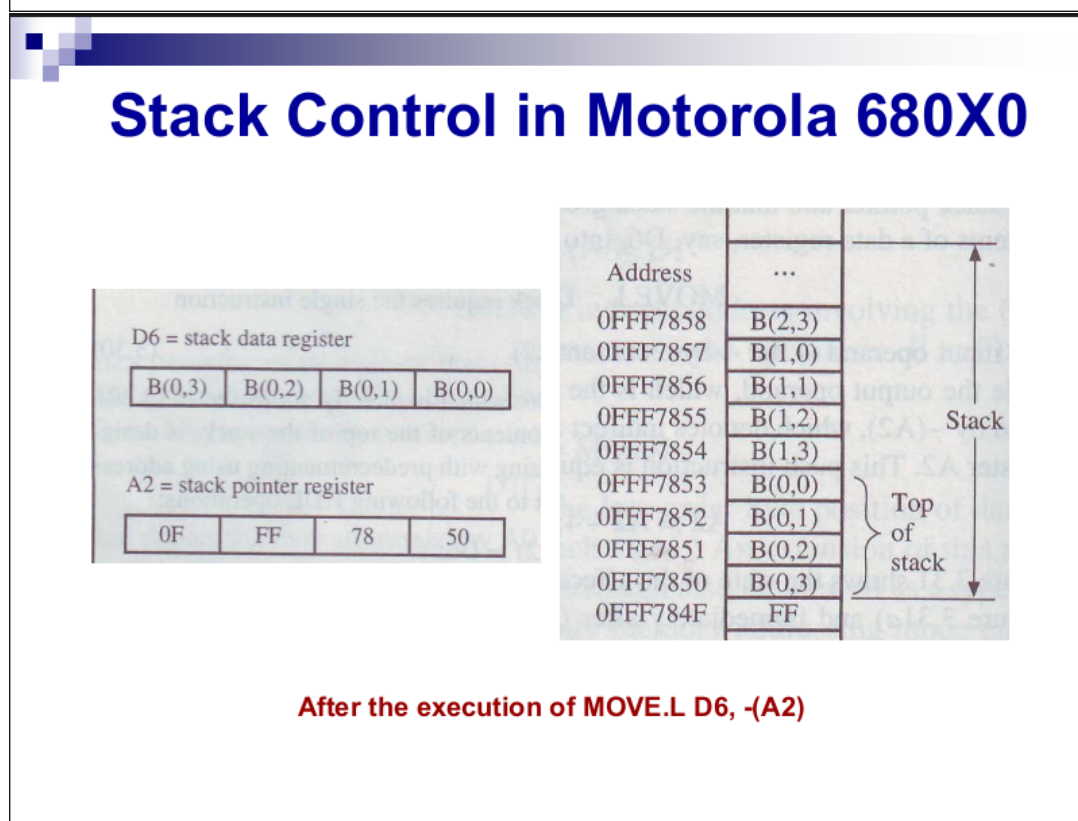
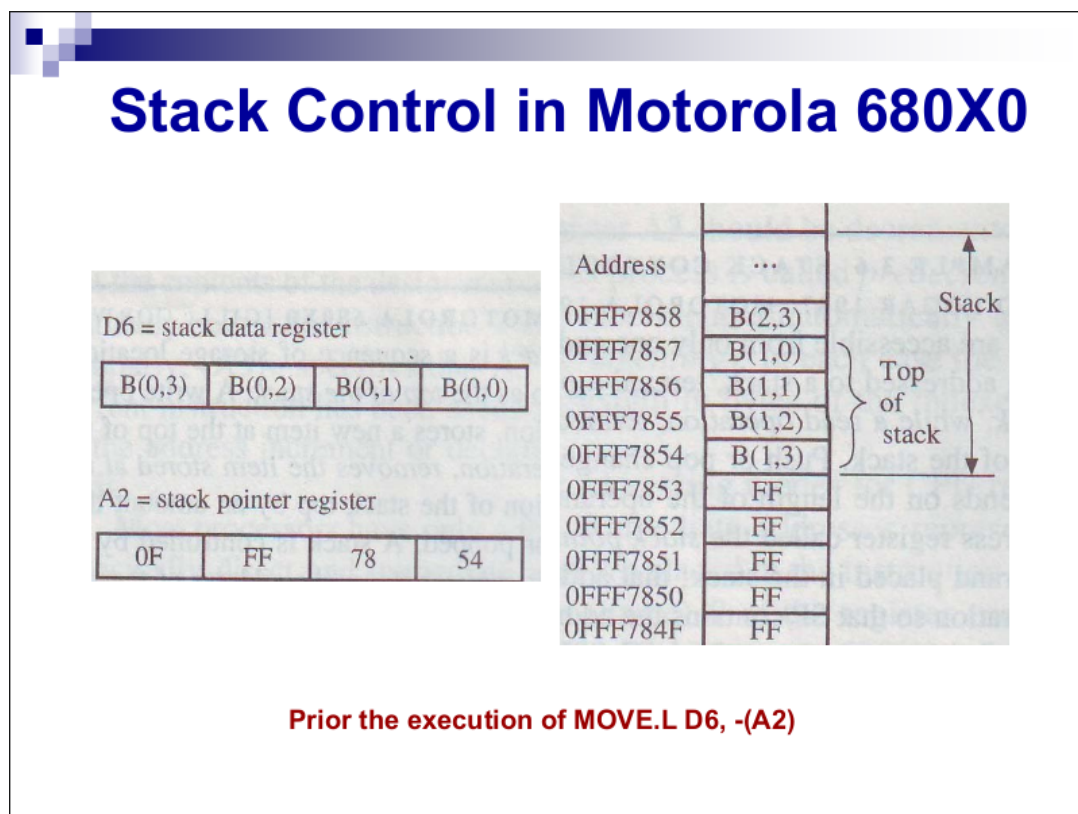
After executing the push:

- $A2 = 0x0FFC$
- Memory at $0x0FFC$ contains the value of D6

After executing the pop:

- D6 is restored from memory at $0x0FFC$
- $A2 = 0x1000$

Diagram (Optional)



(The diagram illustrates the change in A2 and memory contents during push.)