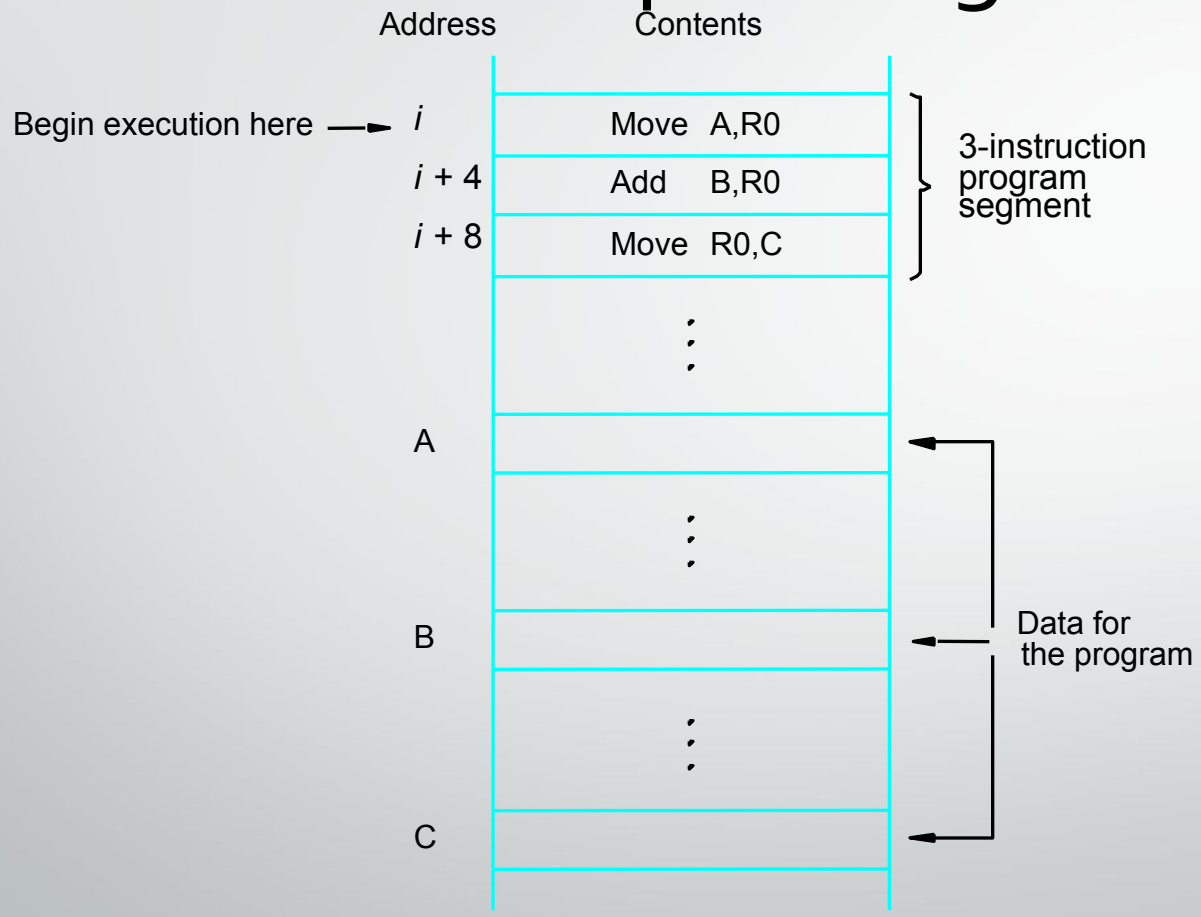
Using Registers

- Registers are faster
- Shorter instructions
  - The number of registers is smaller (e.g. 32 registers need 5 bits)
- Potential speedup
- Minimize the frequency with which data is moved back and forth between the memory and processor registers.

Instruction Execution and Straight-Line Sequencing



Assumptions:
- One memory operand per instruction
- 32-bit word length
- Memory is byte addressable
- Full memory address can be directly specified in a single-word instruction

Two-phase procedure
-Instruction fetch
-Instruction execute

Figure 2.8. A program for C ← [A] + [B].

Branching

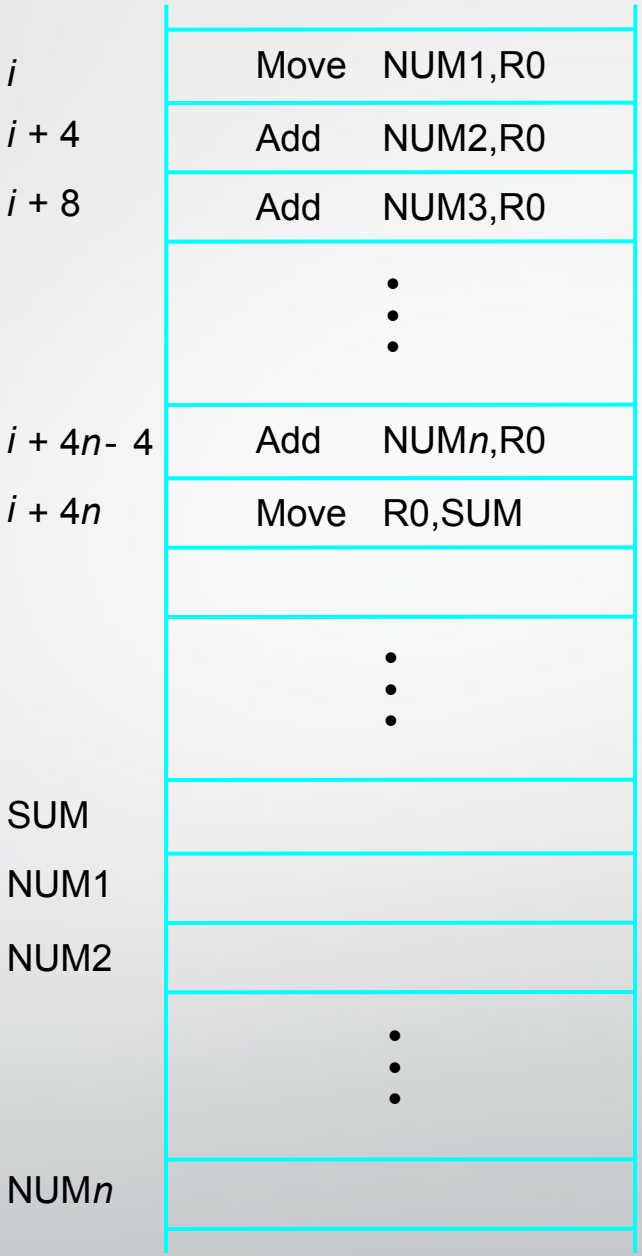


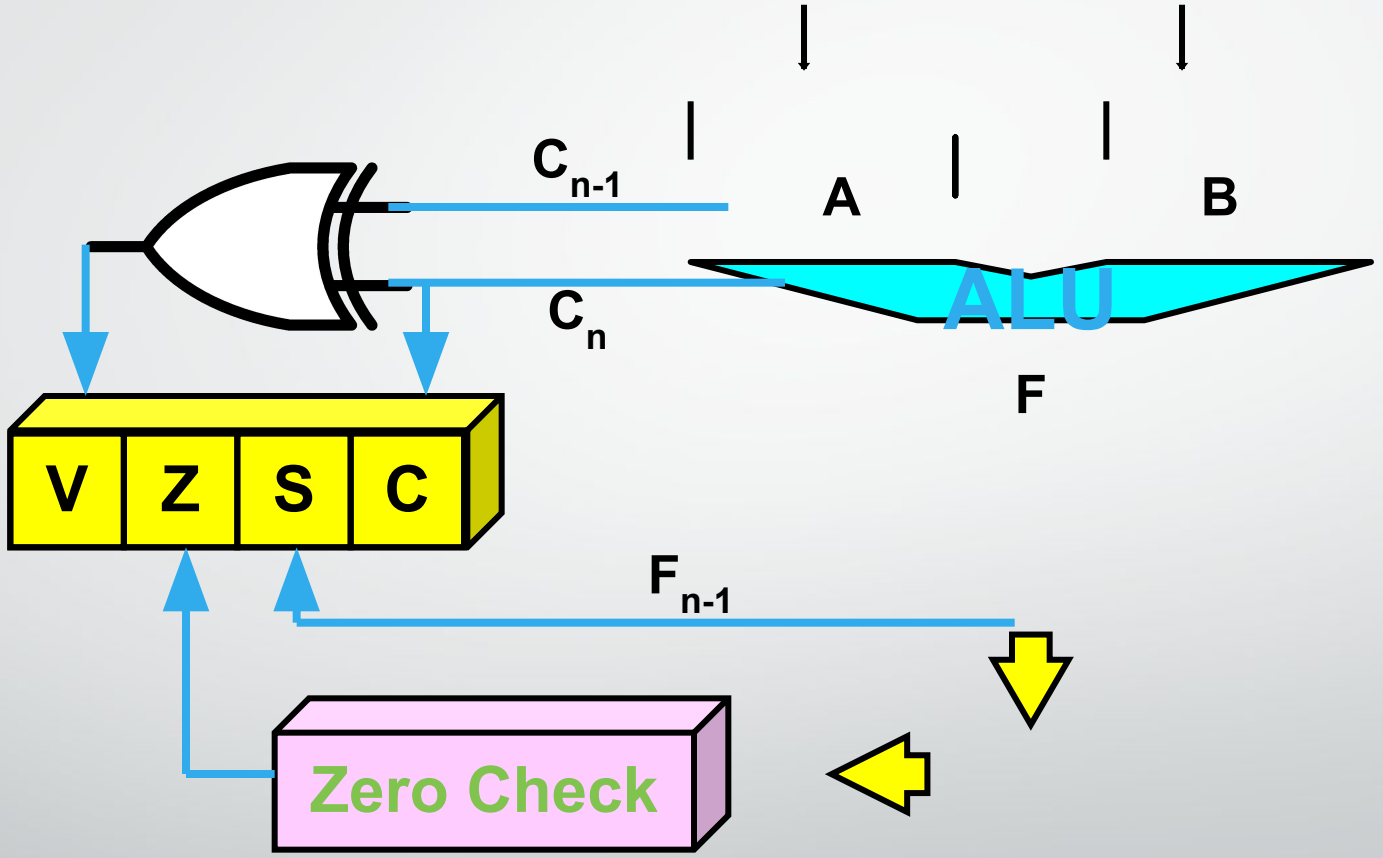
Figure 2.9. A straight-line program for adding *n* numbers.

Condition Codes

- A status register, flag register, or condition code register (CCR) is **a collection of status flag bits for a processor**. Individual bits are implicitly or explicitly read and/or written by the machine code instructions executing on the processor.
- Condition code flags
- Condition code register / status register
- N (negative)
- Z (zero)
- V (overflow)
- C (carry)
- Different instructions affect different flags

- N bit is set if result of operation in negative (MSB = 1)
- Z bit is set if result of operation is zero (All bits = 0)
- V bit is set if operation produced an overflow
- C bit is set if operation produced a carry (borrow on subtraction

Status Bits



Addressing Modes

The addressing modes refers to the way in which the operand of an instruction is specified. The addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually executed.

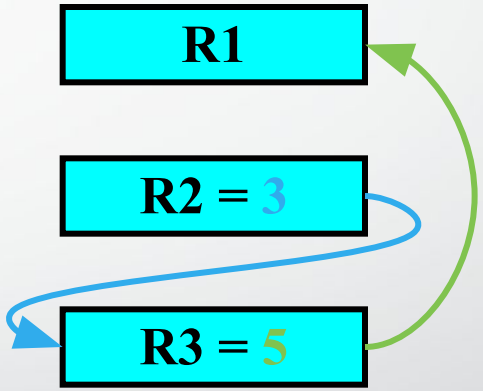
Addressing Modes

- Implied
  - AC is implied in "ADD M[AR]" in "One-Address" instr.
  - TOS is implied in "ADD" in "Zero-Address" instr.
- Immediate
  - The use of a constant in "MOV R1, 5", i.e. R1 ← 5
- Register
  - Indicate which register holds the operand

Addressing Modes

- Register Indirect
  - Indicate the register that holds the number of the register that holds the operand

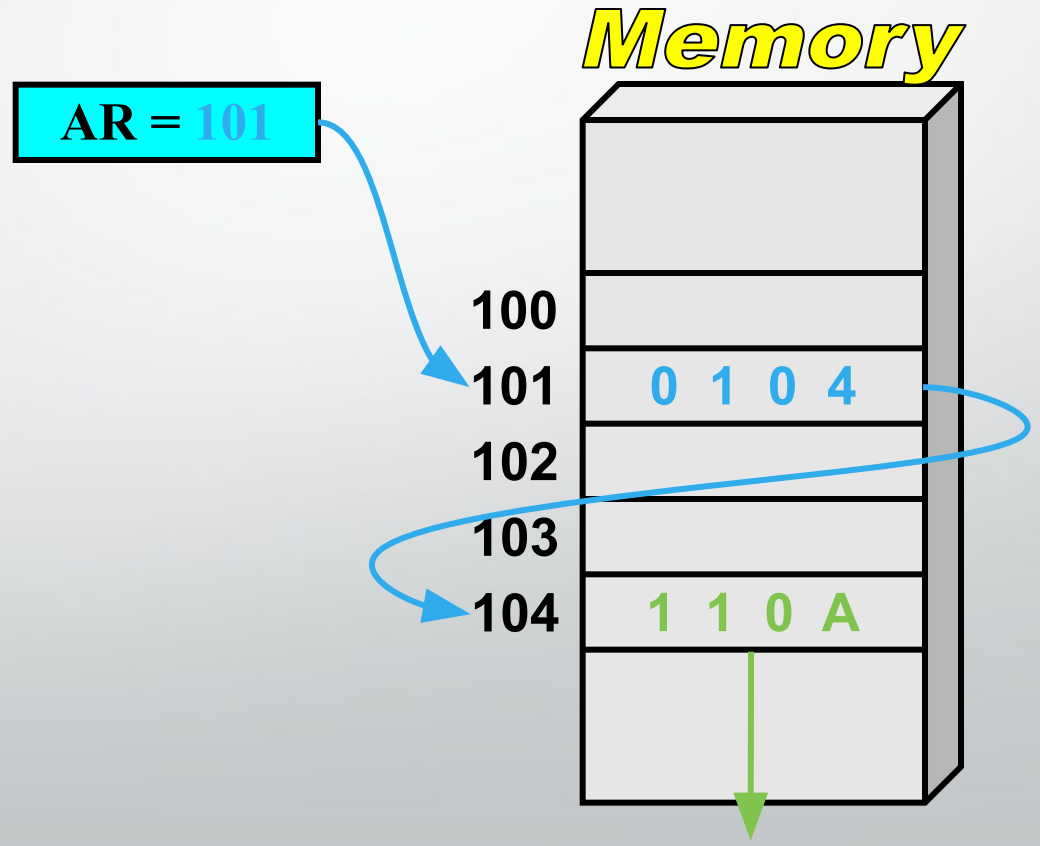
    MOV R1, (R2)



- Autoincrement / Autodecrement
  - Access & update in 1 instr.
- Direct Address
  - Use the given address to access a memory location

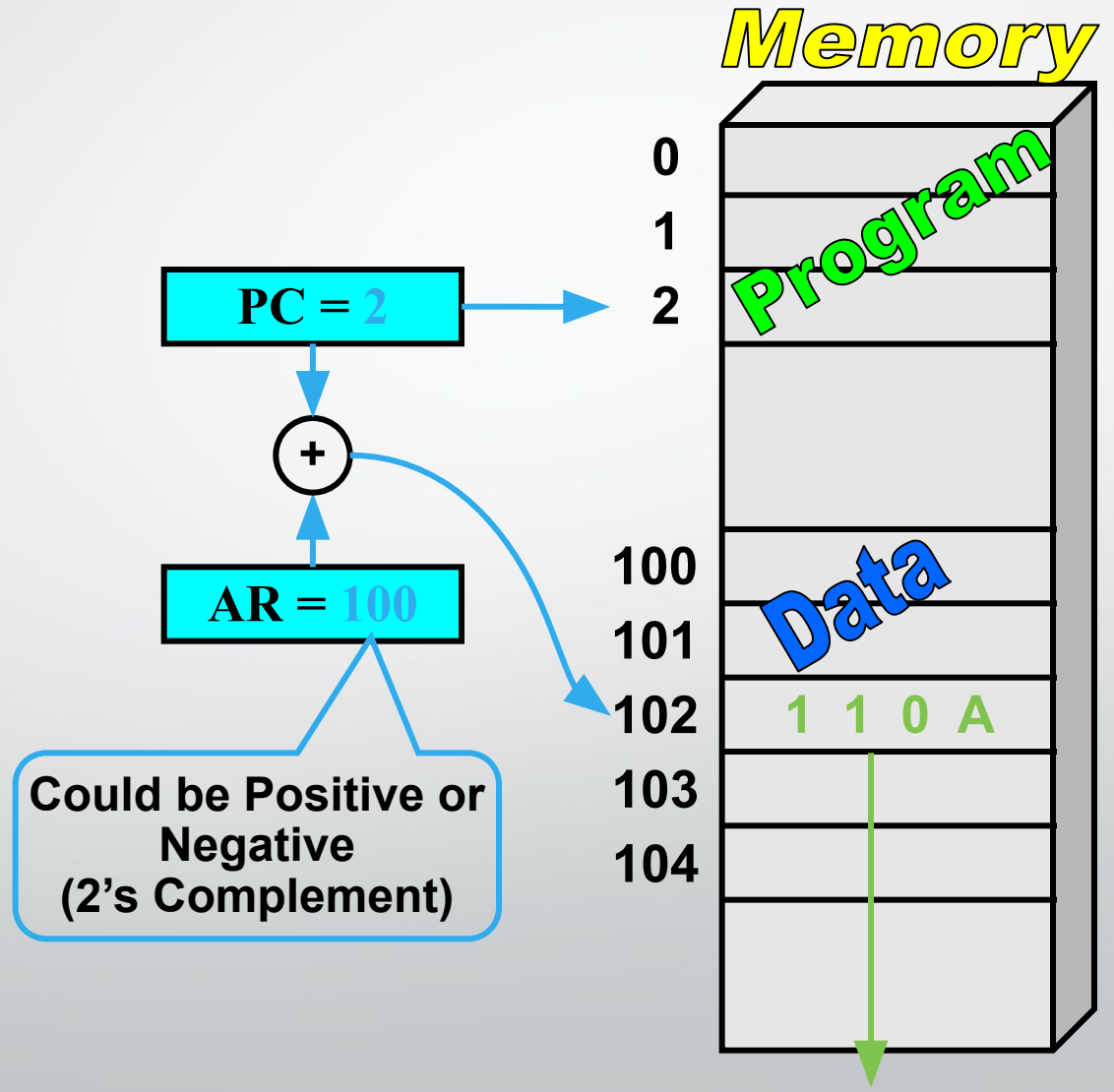
Addressing Modes

- Indirect Address
  - Indicate the memory location that holds the address of the memory location that holds the data



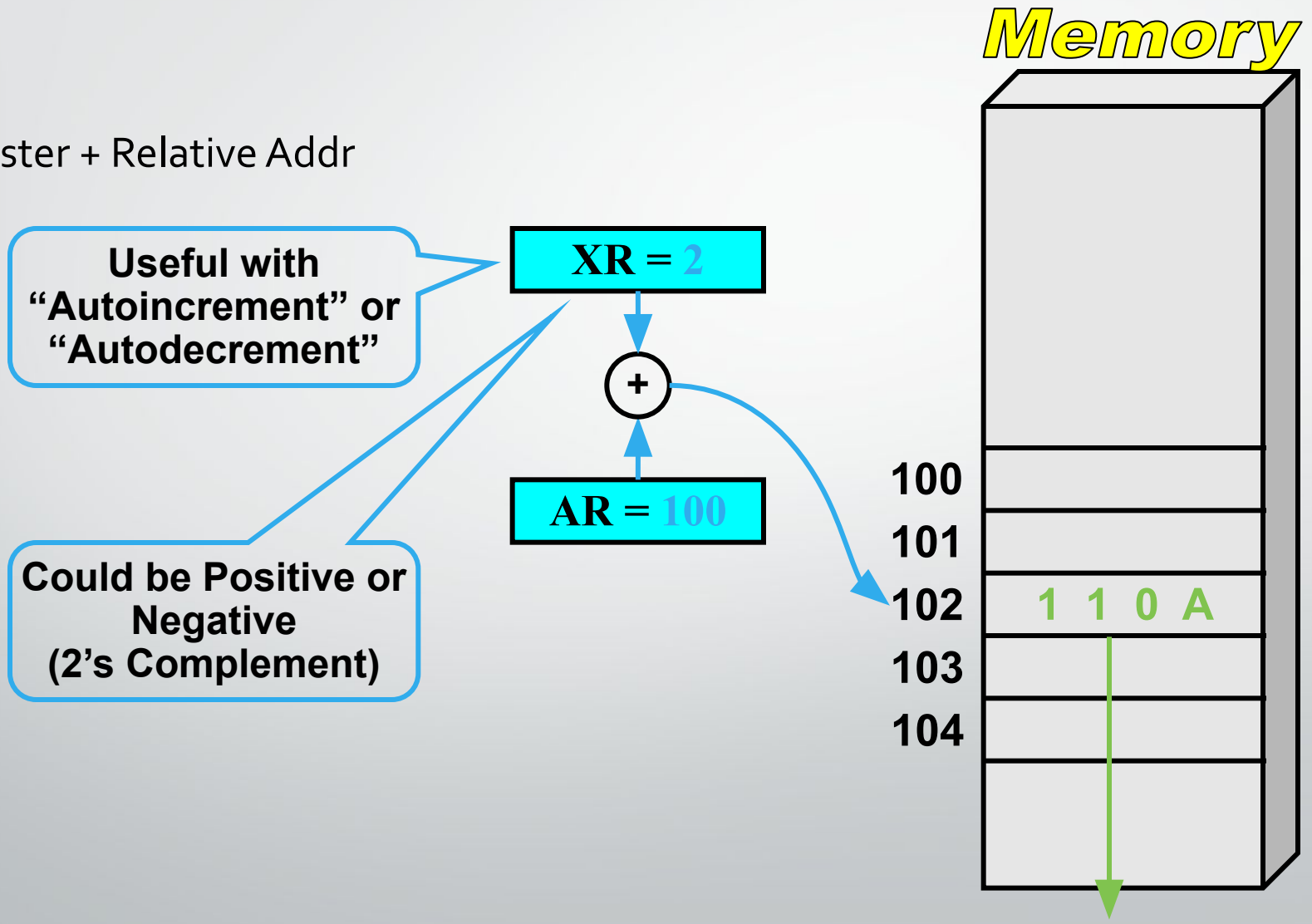
Addressing Modes

- Relative Address
  - *EA* = PC + Relative Addr



Addressing Modes

- Indexed
  - *EA* = Index Register + Relative Addr



Addressing Modes

- Base Register
  - *EA* = Base Register + Relative Addr