

## Addressing Modes

- Immediate
- Direct
- Extended
- Indexed (by X or Y)
- Inherent
- Relative

1

---

---

---


---

---

---

---

---



## Immediate

- LDAA #0x34 (opcode 86, 34)
- LDAB #0xAB (opcode C6, AB)
- Loads a constant
- Structure: opcode constant
- Executes in 2 clock cycles

2

---

---

---


---

---

---

---

---



## Inherent

- Example: INCA (opcode 4C)
- $A = A + 1$
- No additional memory references needed
- Structure: opcode
- INCA executes in 2 cycles

3

---

---

---


---

---

---

---

---



### Direct

- LDAA 0x7E (opcode 96, 7E)
- LDAB 0x34 (opcode D6, 34)
- Loads a zero page memory value
- Structure: opcode, zero page address
- Executes in 3 clock cycles

4

---

---

---


---

---

---

---

---



### Extended

- LDAA 0x1031 (opcode B6, 10, 31)
- LDAB 0x1032 (opcode F6, 10, 32)
- Loads a memory value
- Structure: opcode, 16-bit address
- Executes in 4 cycles

5

---

---

---


---

---

---

---

---



### Indexed

- LDAA 0,x (opcode A6, 00)
- LDAA 12,y (opcode 18, A6, 0C)
- Structure: opcode, offset
- Executes in 4 cycles (x) or 5 cycles (y)
- Memory addressed
  - x + offset or y + offset
  - C++ syntax: x[offset], y[offset]
- Offset is unsigned byte: 0 to 255

6

---

---

---

---

---

---

---

---

### Relative

- Branch instructions
- Branches are relative to the current location, which is the **next** instruction address
- Offset is signed byte
  - Can branch 0x7F (127) bytes forward and 0x80 (128) bytes backward
- Executes in 3 clock cycles

7

---

---

---

---

---

---

---

---

### The bits in the CCR

7 0

S X H I N Z V C    CONDITION CODES

— CARRY/BORROW FROM MSB

— OVERFLOW

— ZERO

— NEGATIVE

— I-INTERRUPT MASK

— HALF CARRY (FROM BIT 3)

— X-INTERRUPT MASK

— STOP DISABLE

---

---

---

---

---

---

---

---

### CCR bits in depth (1/3)

- C – Carry
  - 1 if addition generates a carry
  - 1 if subtraction produces a borrow
  - 1 if shift or rotate shifts out a 1
- V – 2's Complement Overflow
  - 1 if positive + positive = negative
  - 1 if negative + negative = positive
  - Example: 0x7F + 0x50 = 0xCF

9

---

---

---


---

---

---

---

---



### CCR bits in depth (2/3)

- Z – Zero
  - Set if result of last operation is 0 (if all bits are 0)
- N – Negative
  - Set if MSB (7 or 15) of result is 1
- I – Interrupt mask
  - Masks (defers) certain interrupts when set
- H – Half carry
  - Set if an 8-bit addition causes a carry from lower to upper nibble (useful for nibble operations)

10

---

---

---


---

---

---

---

---



### CCR bits in depth (3/3)

- X – Interrupt mask
  - Set (1) at power-up
  - Once cleared cannot be set except by reset
  - When clear,  $\sim$ XIRQ is pin recognized
- S – Stop disable
  - If 1, STOP is ignored (operates as a NOP)

11

---

---

---

---

---

---

---

---