Addressing Modes:

Inherent Addressing: Only between resisters

Example: Transfer A to B

Sum A and B into A or B

Immediate Addressing: DATA after opcode

Example: LDAA # Data 8bit - Load A w/ immediate data

LDX # Data 16Bit - Load X w/ immediate data

Data is put directly into the A or B accumulators.

Data is loaded byte-by-byte low first then high into X or y

Other: ADDA # Data 8Bit - Add A w/ immediate data

But what about this one? How could we do this? write over B with #Data and that won't be good. We could just store B to data memory, load the immediate data, add it, then restore B. Or another way could be to add an invisible reg.

Extended Addressing: Address After opcode

Example: LDAA Addr 16 Bit - Load A with data from memory location.

Since we are loading a 16 Bit address, and we don't want to overwrite X or Y, We load the address into the Memory Address Register (MAR) low byte first, then high byte. Once in MAR, we switch the Address Bus Mux to select MAR as the address source, and then load the corresponding data output from the memory into A or B.

Other: ADDA ADDR 16Bit Add A to data pointed to by addr, and store in A.

How could we do this?

Indexed Addressing: Pointer With offset

8 bit displacement

Example: LDAA dd, X — Load A with data pointed to by X+dd

In this configuration, the opcode will be followed by the eight bit offset which will be loaded into an 8 bit displacement register, which is contained near the x and y registers. This is then added to X through a series of chained adders, and then used to pull data from memory once the Address Select Mux has been configured to select X or y as the address source.

Absolute Addressing: Modify the program counter

Example: BNE addrL ← 8 bit ~~offset~~ Branch if Z=1 to address pointed to by

PC = (PC & 0xFF00) | (addrL & 0xFF

With these instructions, we are basically changing the program counter directly to suit our branching needs. We change the PC if the branch condition is true, and we do it by simply loading the 8 bit value after the opcode into the low byte of the program counter.