


Logical Instructions

- Logical ANDs
 - Set/clear N/Z, V to 0
- ANDA/ANDB – use register A or B
 - `reg = reg & mem; // &=`
 - Imm, ext, dir, indexed
- BITA/BITB – bit test A or B
 - Sets CCR like AND_
 - But A/B unmodified
 - Same addressing modes as AND_


1



Logical Instructions

- EORA/EORB – exclusive or A/B with “memory”
 - `reg = reg ^ mem; // ^=`
 - Imm, ext, dir, indexed
 - Set/clear N/Z, V to 0
- ORAA/ORAB – or A/B with “memory”


2



Logical Instructions

- Unlike earlier ones, these affect **memory**
- BCLR – bit clear
 - 1 in mask clears
 - direct and indexed only
 - like AND with inverted mask
- BSET – bit set
 - 1 in mask sets
 - like OR with mask


3



BSET and BCLR

- Assembler syntax is a little different
- Modes: indexed and direct
- Also supply a mask
- Examples:
 - `bset 2,x #0b11110000 ; idx, sets 4 high bits`
 - `bclr *0xB0 #0b00111100 ; dir, clears 4 mid. bits`
- No extended mode
- Use AND_, OR_ instead for accs. A & B


4



Bit changes

- Most often need to set, clear, or flip an individual bit
- Set – OR with 1
- Clear – AND with 0
- Flip – exclusive or (EOR) with 1


5



Branches

- Branch types
 - Simple (unconditional)
 - Always, Never, Subroutine
 - Simple conditional
 - 1 CCR bit
 - Signed conditional
 - 1 or more CCR bits representing signed results
 - Unsigned conditional
 - 1 or more CCR bits representing unsigned results
 - Bit test
 - If memory bits set/clear, no CCR use


6



Branches – characteristics

- All branches have a distance limit
- Branches are relative to
 - Current location
 - Which equals address of next instruction
- Program can branch up to
 - 0x7F (127) bytes (forward)
 - 0x80 (-128) bytes (backward)


7



Simple Branches

- BRA – branch always
- BSR – branch to subroutine
- BRN – branch never (a 3 cycle NOP)


8



Simple Conditional Branches

- BMI – branch on minus (if N=1)
- BPL – branch on plus (if N=0)
- BEQ – branch on equal (if Z=1)
- BNE – branch on not equal (if Z=0)
- BVS – branch on overflow set (if V=1)
- BVC – branch on overflow clear (if V=0)
- BCS – branch on carry set (if C=1)
- BCC – branch on carry clear (if C=0)


9



Signed Conditional Branches

- BEQ – branch if equal
- BNE – branch if not equal
- BGT – branch if greater than
- BLT – branch if less than
- BLE – branch if less than or equal
- BGE – branch if greater than or equal


10



Unsigned Conditional Branches

- BEQ – branch if equal
- BNE – branch if not equal
- BHI – branch if higher
- BLO/BCS – branch if lower
- BHS/BCC – branch if higher or the same
- BLS – branch if lower or the same


11



Branches

- OK, but how are the CCR flags set?
- Comparisons!


12



Setting the flags – 8-bit compares

- CBA – compare B to A (does a A-B, but just sets flags) – flags set NZVC
- CMPA – compare A to memory (A-m)
 - Modes: immediate, direct, ext., indexed
 - Sets: NZVC
- CMPB – compare B to memory (B-m)
 - Modes: immediate, direct, ext., indexed
 - Sets: NZVC


13



Setting the flags – 16-bit compares

- CPD – compare D to memory
 - Set flags for D-[m:m+1]
 - Modes: immediate, direct, ext., indexed
 - Sets: NZVC
- CPX – compare X to memory
- CPY – compare Y to memory


14



Bit test branches

- brclr 4,x 0b00111100 L2
 - Branch if the 4 middle bits at 4,x are clear
- brset *0xB2 0b01000000 L1
 - Branch if the 2nd MSB at 0x00B2 is set
- No effect on memory or CCR
- Indexed and direct modes only
 - Branch offset is relative

15



Programming Exercises

- Hex to ASCII
- Average n numbers
- Table look-up

16
