EE4390 Microprocessors

Lesson 4
Programming Model, Assembly Language, Instruction Execution Cycle
Overview

• Programming Model
• Motorola Assembly Language
• Instruction Execution Cycle
# Programming Model

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th></th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>7</td>
<td>B</td>
<td>7</td>
</tr>
</tbody>
</table>

Two 8-bit accumulators

or

<table>
<thead>
<tr>
<th>15</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>0</td>
</tr>
</tbody>
</table>

One 16-bit accumulator

<table>
<thead>
<tr>
<th>15</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>IX</td>
<td>0</td>
</tr>
</tbody>
</table>

16-bit index register X

<table>
<thead>
<tr>
<th>15</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>IY</td>
<td>0</td>
</tr>
</tbody>
</table>

16-bit index register Y

<table>
<thead>
<tr>
<th>15</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>0</td>
</tr>
</tbody>
</table>

16-bit stack pointer

<table>
<thead>
<tr>
<th>15</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>0</td>
</tr>
</tbody>
</table>

16-bit program counter

<table>
<thead>
<tr>
<th>7</th>
<th>SXEHINZVC</th>
<th>0</th>
</tr>
</thead>
</table>

Condition Code Register

Revised: Aug 1, 2003
Programming Model (cont)

• A, B: 8-bit accumulators
  – collectively D: 16-bit register
• X, Y: 16-bit index registers
• SP: 16-bit stack pointer
• PC: 16-bit program counter
• CCR: 8-bit Condition Code Register
Programming Model (cont)

- Condition Code Register - Reference Appx A

![Diagram of Condition Code Register]

Revised: Aug 1, 2005
Programming Model (cont)

- S: disables STOP instruction
- X: enables nonmaskable interrupt
- H: half carry flag from lower nibble to upper nibble
- I: enable maskable interrupts
- N: negative flag
- Z: zero flag
- V: overflow flag
- C: carry flag

Flag activities:
1 : sets
0 : resets
- : no change
△ : determined by operation
Motorola Assembly Language

<table>
<thead>
<tr>
<th>Label</th>
<th>Op-Code</th>
<th>Operand(s)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>start:</td>
<td>LDAA</td>
<td>#FC</td>
<td>;load acc A</td>
</tr>
</tbody>
</table>

- Label: name for a memory location
- Op-Code: mnemonic, action part of instruction
- Operand: access to data for instruction via addressing mode
- Comment: “;” indicates comment follows
  - can also be used at the beginning of a line
Assembly vs C

• Assembly:
  – better HW control, faster

• C:
  – more readable
  – Top-Down Design
  – more efficient programming
  – do not need to know HW details
Instruction Execution Cycle

- Fetch
- Decode
- Execute

Revised: Aug 1, 2003
“The Big Picture”

ICC12 - host PC

filename.c
filename.h

filename.asm

filename.asm

filename.asm

filename.lst
filename.s19

comm link via SCI

A4 EVB
• HW
• SW
• DEBUG12

daughter card

fab area
ICC12 Specific Items

.area name(abs)

.org $1000

_main::

loop::

;sample program
Sample Code

; File Name: introlab.s
; File Created: 04-08-02
; File Modified: 04-13-02
; Author(s): Abbie Wells
; Introductory Lab Exercise to demonstrate uses of the
; HC12 Teaching Platform and familiarize the user with
; assembly code. Basic arithmetic operations will be
; performed, results and CCR contents stored to memory,
; and CCR contents displayed to the LEDs.
.area introlab(abs)
.org $4100
_main::

    PORTT = $00AE   ; Create label for Port T
    DDRT = $00AF    ; Create label for DDRT
    TSCR = $0086    ; Create label for TSCR

; Initialize Port T to be an output port
## Memory Map - A4 Evaluation Board

### Table 3-5. Factory-Configuration Memory Map

<table>
<thead>
<tr>
<th>Address Range</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0000 - $01FF</td>
<td>CPU registers</td>
<td>on-chip (MCU)</td>
</tr>
<tr>
<td>$0800 - $09FF</td>
<td>user code/data</td>
<td>1K on-chip RAM (MCU)</td>
</tr>
<tr>
<td></td>
<td>reserved for D-Bug12</td>
<td></td>
</tr>
<tr>
<td>$1000 - $1FFF</td>
<td>user code/data</td>
<td>4K on-chip EEPROM (MCU)</td>
</tr>
<tr>
<td>$4000 - $7FFF</td>
<td>user code/data</td>
<td>16K external RAM (U4, U6A)</td>
</tr>
<tr>
<td>$8000 - $9FFF</td>
<td>available for user programs*</td>
<td>32K external EPROM (U7, U9A)</td>
</tr>
<tr>
<td>$A000 - $FD7F</td>
<td>D-Bug12 program</td>
<td></td>
</tr>
<tr>
<td>$FD80 - $FDFF</td>
<td>D-Bug12 startup code*</td>
<td></td>
</tr>
<tr>
<td>$FE00 - $FE7F</td>
<td>user-accessible functions</td>
<td></td>
</tr>
<tr>
<td>$FE80 - $FEFF</td>
<td>D-Bug12 customization data*</td>
<td></td>
</tr>
<tr>
<td>$FF00 - $FF7F</td>
<td>available for user programs*</td>
<td></td>
</tr>
<tr>
<td>$FF80 - $FFFF</td>
<td>reserved for interrupt and reset vectors</td>
<td></td>
</tr>
</tbody>
</table>

*Code in these areas may be modified. Requires reprogramming of the EPROMs—refer to Appendix E Customizing the EPROMs.

“Copyright of Motorola, Used by Permission”