EECE 276
Embedded Systems

HC 12 Overview: Addressing modes, instructions
HC12 Introduction

- "Big brother" of HC11
- Source code compatibility
  » Binary code is NOT compatible!
- Similar programmer’s model
  » (A/B-D,X,Y,SP,PC)
- 16-bit architecture – 16 bit datapath
- Instruction queue
  » Pipelined cache to speed up processing
- Improved addressing modes
- New instructions
HC12 Programmer’s Model

8-BIT ACCUMULATORS A AND B
16-BIT DOUBLE ACCUMULATOR D
INDEX REGISTER X
INDEX REGISTER Y
STACK POINTER
CONDITION CODE REGISTER
- CARRY
- OVERFLOW
- ZERO
- NEGATIVE
- MASK (DISABLE) IRQ INTERRUPTS
- HALF-CARRY (USED IN BCD ARITHMETIC)
- MASK (DISABLE) XIRQ INTERRUPTS
- RESET OR XIRQ SET X, INSTRUCTIONS MAY CLEAR X BUT CANNOT SET X
- STOP DISABLE (IGNORE STOP OPCODES) RESET DEFAULT IS 1

Stack before/after IT

Note: Different from HC11 – SP points to last element pushed, NOT next empty slot
## HC12 New Addressing Modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Instruction</th>
<th>Mode</th>
<th>5-bit signed constant offset from X, Y, SP, or PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indexed (5-bit offset)</td>
<td>INST oprx5,xyz</td>
<td>IDX</td>
<td>Auto pre-decrement x, y, or sp by 1 ~ 8</td>
</tr>
<tr>
<td>Indexed (pre-decrement)</td>
<td>INST oprx3,-xyz</td>
<td>IDX</td>
<td>Auto pre-decrement x, y, or sp by 1 ~ 8</td>
</tr>
<tr>
<td>Indexed (pre-increment)</td>
<td>INST oprx3,+xyz</td>
<td>IDX</td>
<td>Auto pre-increment x, y, or sp by 1 ~ 8</td>
</tr>
<tr>
<td>Indexed (post-decrement)</td>
<td>INST oprx3,xyz-</td>
<td>IDX</td>
<td>Auto post-decrement x, y, or sp by 1 ~ 8</td>
</tr>
<tr>
<td>Indexed (post-increment)</td>
<td>INST oprx3,xyz+</td>
<td>IDX</td>
<td>Auto post-increment x, y, or sp by 1 ~ 8</td>
</tr>
<tr>
<td>Indexed (accumulator offset)</td>
<td>INST abd,xyz</td>
<td>IDX</td>
<td>Indexed with 8-bit (A or B) or 16-bit (D) accumulator offset from X, Y, SP, or PC</td>
</tr>
<tr>
<td>Indexed (0-bit offset)</td>
<td>INST oprx9,xyz</td>
<td>IDX1</td>
<td>9-bit signed constant offset from X, Y, SP, or PC (lower 8 bits of offset in one extension byte)</td>
</tr>
<tr>
<td>Indexed (16-bit offset)</td>
<td>INST oprx16,xyz</td>
<td>IDX2</td>
<td>16-bit constant offset from X, Y, SP, or PC (16-bit offset in two extension bytes)</td>
</tr>
<tr>
<td>Indexed-Indirect (16-bit offset)</td>
<td>INST [oprx16,xyz]</td>
<td>[IDX2]</td>
<td>Pointer to operand is found at...16-bit constant offset from X, Y, SP, or PC (16-bit offset in two extension bytes)</td>
</tr>
<tr>
<td>Indexed-Indirect (D accumulator offset)</td>
<td>INST [D,xyz]</td>
<td>[D,IDX]</td>
<td>Pointer to operand is found at...X, Y, SP, or PC plus the value in D</td>
</tr>
</tbody>
</table>
HC12 Addressing Modes

- INH, IMM, DIR, EXT, REL as before
- Improved indexing:
  » X, Y, SP(PC) can all be used
- IDX variants:
  » Constant offset:
    - 5-bit signed constant offset, X/Y/SP/PC
    - 9-bit signed constant offset, X/Y/SP/PC
    - 16-bit constant offset, X/Y/SP/PC
  » Accumulator offset:
    - A/B(8-bit) or D(16-bit) offset, X/Y/SP/PC
HC12 Addressing Modes

Improved indexing

» Automatic pre/post decrement/increment, X/Y/SP:
  – Pre: Before accessing data, Post: after accessing data
  – Increment/Decrement by 1~8

<table>
<thead>
<tr>
<th>HC11</th>
<th>HC12</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSHA</td>
<td>STAA 1,-SP</td>
</tr>
<tr>
<td>PULA</td>
<td>LDAA 1,SP</td>
</tr>
<tr>
<td>PSHX</td>
<td>STX 2,-SP</td>
</tr>
<tr>
<td>PULX</td>
<td>LDX 2,SP+</td>
</tr>
</tbody>
</table>

– Example:
  • In C: while(*x++ = *y++);

L
LDAA 1,Y+
STAA 1,X+
BNE L
NEXT
HC12 Addressing Modes

Improved indexing

- Indexed-indirect
  INST [oprx16,xy邵]
  Pointer to operand is found at 16-bit constant offset from X/Y/SP/PC

- Indexed-indirect
  INST [D,xy邵]
  Pointer to operand is found at X/Y/SP/PC plus the value in D

Rationale for new addressing modes:
Better support for high-level languages/compilers
New HC12 Instructions

- Direct Memory-to-memory move
  
  LOOP MOVW 2,X+ , 2,Y+ Move word, incr. X/Y
  
  DBNE B,LOOP Repeat B times

- Universal transfer instruction
  
  » Exchange between any two registers

- Loop constructs:
  
  – Decrement/Increment/Test counter, then branch on zero/non-zero: D/I/T BNE/BEQ

- Minimum/maximum instructions:
  
  – MINA op: set A to min of (A,op) - also MAXA
  – MINM op: set M to min of (A,M) - also MAXM
New HC12 Instructions

- Fuzzy Logic Instructions
  - Special instructions for fuzzy logic calculations
- Table lookup instructions: TBL, ETBL
  - Linear interpolation in tables with 8/16 bit values
  - B: fractional value (<1!), opr: points to y1
  - Result = y1 + B * (y2-y1)
- Bit manipulations instructions work in direct/extended/indexed addressing modes
- Push/pull D and CCR (3 bytes)
- Compare SP value

For details, see:

CPU12 Reference Guide
Transporting MC68HC11 Code to MC68HC12 Devices
Paged Memory in HC12

9S12DP256 has 256kbytes of flash memory on chip. It is divided into 16x16 kbyte pages, which are accessible at $8000.

Which actual page from the 16 is active, is determined by a page register: PPAGE

CPU still uses 16-bit addresses. If you want to use pages, the PPAGE register must be explicitly manipulated.

Special instructions:

```
CALL addr, page
RTC
```