Computer Science CSCI 261

Computer Architecture and Assembly Language

Dr. Peter Walsh

Department of Computer Science

Vancouver Island University

peter.walsh@viu.ca

Motorola 68HC11

The 68HC11 (6811 or HC11 for short) is an 8-bit microcontroller family introduced by Motorola in 1984. HC11s are used in automotive applications, barcode readers, hotel card key writers, amateur robotics, and various other embedded systems [Wikipedia].

◯ E Series Development Boards

- HC11 EVBU (711E9)

- AXIOM CME11E9 EVBU (11E9)

Development Environments

○ MS Windows

- `as11`: cross assembler
- `ImageCraft`: cross compiler
- `sim11`, `wookie`: simulators
- `HyperTerminal`: down-loader/communications

○ Linux

- `gas`: cross assembler (GNU Tool Chain for HC11)
- `gcc`: cross compiler (GNU Tool Chain for HC11)
- `gdb`: simulator (GNU Tool Chain for HC11)
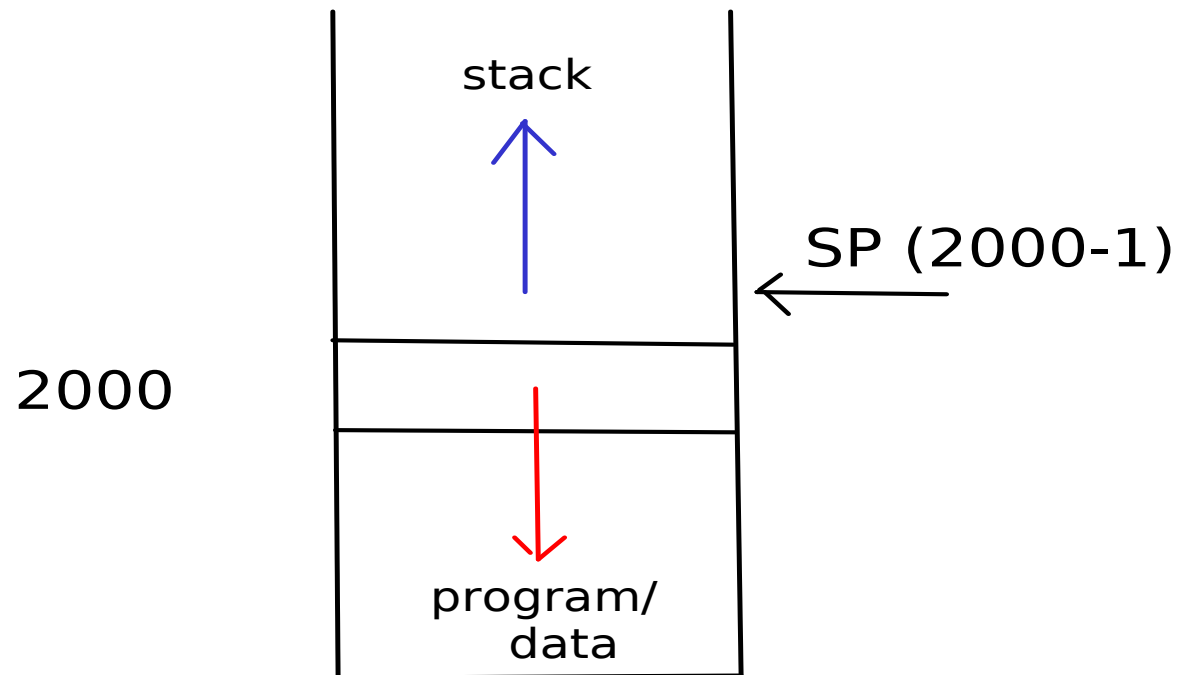- `minicom`: down-loader/communications

HC11 Overview

○ Components

- cpu
- five ports `A`, `B`, `C`, `D` and `E`
- serial peripheral interface (SPI)
- serial communications interface (SCI)
- timer subsystem (Tic/Toc)
- analog to digital subsystem (A/D)
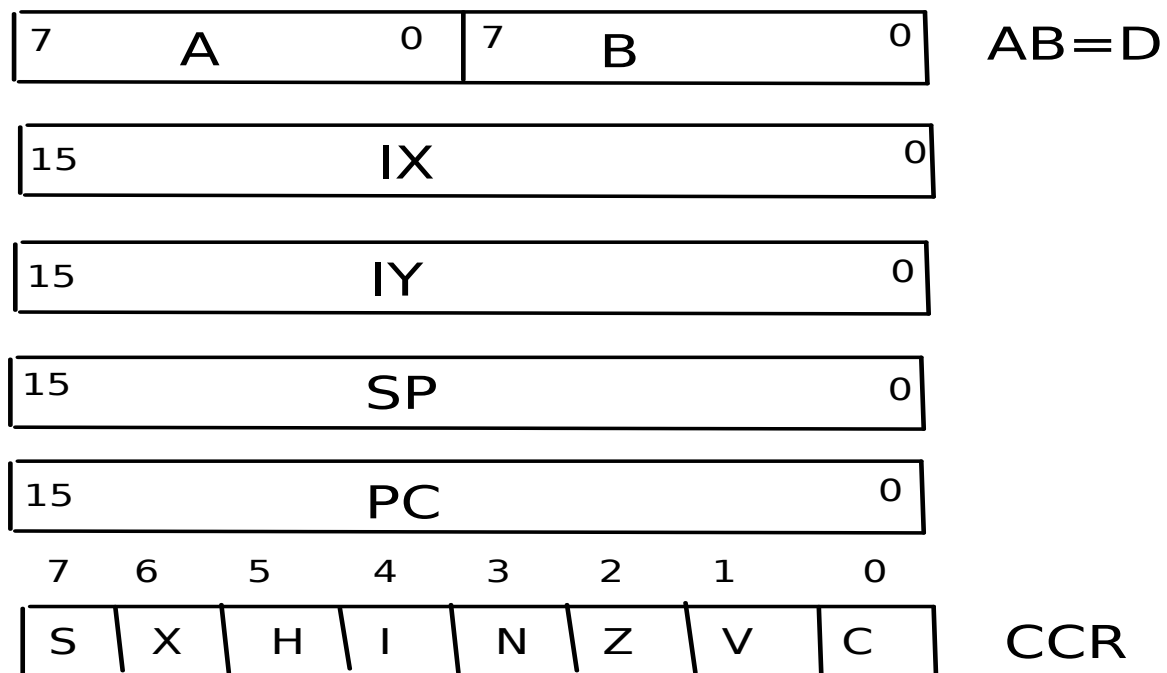- interrupt subsystem
- mode select

HC11 Modes

○ Single Chip

○ Expanded
- Ports `B` and `C` are used
  to access external memory
- Ports `D` and `A` and `E` are available
  for input/output

○ Boot Strap

○ Test

EVBU Stack

stack

SP (2000-1)

2000

program/
data

HC11 Architecture

| 7 | A | 0 | 7 | B | 0 | AB=D |

| 15 | IX | 0 |

| 15 | IY | 0 |

| 15 | SP | 0 |

| 15 | PC | 0 |

7	6	5	4	3	2	1	0
S	X	H	I	N	Z	V	C

CCR

HC11 Condition Code Register (CCR)

○ Flags

- `C` carry
- `V` overflow
- `Z` zero
- `N` negative
- `I` interrupt mask
- `H` half carry (from bit 3)
- `X` X interrupt mask
- `S` stop disable

HC11 Assembly Language Programming

○ Assemble Statement Types
- instructions
- directives

Assembly Language Directives are instructions to the assembler to help manage the assembly process. The assemblers `as11` and `gas` target the same assembly language but use different directives. Our text book uses `asm11` directives. We will use `gas` directives in our code examples.