Fig 1.3 Programmer’s Models of 4 Commercial Machines

M6800 (introduced 1975)
- 7 0
  A B
  15 IX SP
  6 special purpose registers
  PC Status

18086 (introduced 1979)
- 15 8 7 0
  AX BX CX DX
  Data registers
  SP BP SI DI
  Address and count registers

VAX11 (introduced 1981)
- 31 0
  12 general purpose registers
  Rp R11 AP
  FP SP PC
  PSW

PPC601 (introduced 1993)
- 32 63
  32 64-bit floating point registers
  0 31
  32 32-bit general purpose registers
  0 31
  More than 50-32-bit special purpose registers
  0 31
  2^32 bytes of main memory capacity

M6800:
- 2^16 bytes of main memory capacity
- Fewer than 100 instructions
- 2^16 - 1

18086:
- Memory segment registers
- 2^20 bytes of main memory capacity
- More than 120 instructions
- 2^20 - 1

VAX11:
- 2^32 bytes of main memory capacity
- More than 300 instructions

PPC601:
- 2^52 bytes of main memory capacity
- More than 250 instructions
Machine, Processor, and Memory State

- The Machine State: contents of all registers in system, accessible to programmer or not
- The Processor State: registers internal to the CPU
- The Memory State: contents of registers in the memory system
- “State” is used in the formal finite state machine sense
- Maintaining or restoring the machine and processor state is important to many operations, especially procedure calls and interrupts
Data Type: HLL Versus Machine Language

- HLLs provide type checking
  - Verifies proper use of variables at compile time
  - Allows compiler to determine memory requirements
  - Helps detect bad programming practices
- Most machines have no type checking
  - The machine sees only strings of bits
  - Instructions interpret the strings as a type: usually limited to signed or unsigned integers and FP numbers
  - A given 32-bit word might be an instruction, an integer, a FP number, or 4 ASCII characters
### Tbl 1.3 Instruction Classes

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Class</th>
<th>C</th>
<th>VAX</th>
<th>Assembly</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Movement</td>
<td>a = b</td>
<td>MOV b, a</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arithmetic/logic</td>
<td>b = c + d*e</td>
<td>MPY d, e, b</td>
<td>ADD c, b, b</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control flow</td>
<td>goto LBL</td>
<td>BR LBL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- This compiler:
  - Maps C integers to 32-bit VAX integers
  - Maps C assign, *, and + to VAX MOV, MPY, and ADD
  - Maps C goto to VAX BR instruction
- The compiler writer must develop this mapping for each language-machine pair
Tools of the Assembly Language Programmer’s Trade

- The assembler
- The linker
- The debugger or monitor
- The development system
Who Uses Assembly Language

- The machine designer
  - Must implement and trade off instruction functionality
- The compiler writer
  - Must generate machine language from a HLL
- The writer of time or space critical code
  - Performance goals may force program-specific optimizations of the assembly language
- Special purpose or imbedded processor programmers
  - Special functions and heavy dependence on unique I/O devices can make HLLs useless