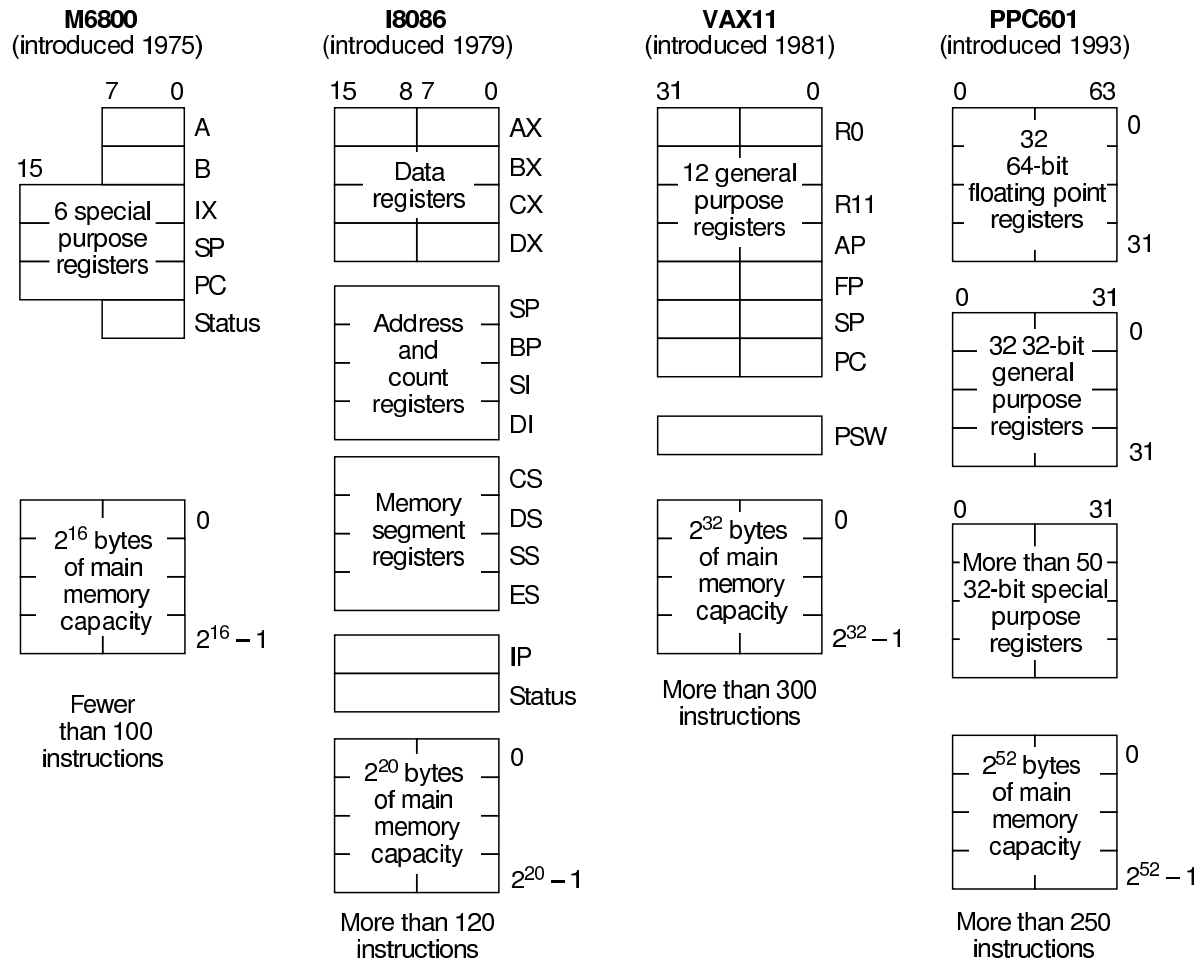


# Fig 1.3 Programmer's Models of 4 Commercial Machines



# 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

Instruction Class	C	VAX Assembly Language
Data Movement	$a = b$	MOV b, a
Arithmetic/logic	$b = c + d * e$	MPY d, e, b ADD c, b, b
Control flow	goto LBL	BR LBL

- **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**