



## An Example: printword.mli

```

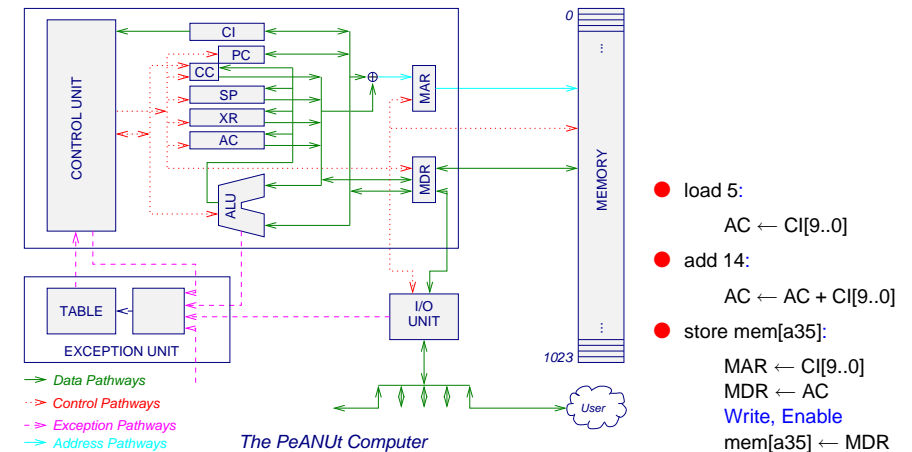
; Simple example machine language program to print a word

START a10                ; start address of program, init. PC to this

AT a10                   ; store the following data items (instrns)
                          ; into memory from a10 onwards
000 001 0 001 001 000   ; a10: load 'H' (immediate mode)
110101 0 000 000 011   ; a11: trap 3 (put)
000 001 0 001 000 101   ; a12: load 'E' (immediate mode)
110101 0 000 000 011   ; a13: trap 3 (put)
000 001 0 001 001 100   ; a14: load 'L' (immediate mode)
110101 0 000 000 011   ; a15: trap 3 (put)
000 001 0 001 001 100   ; a16: load 'L' (immediate mode)
110101 0 000 000 011   ; a17: trap 3 (put)
000 001 0 001 001 111   ; a20: load '0' (immediate mode)
110101 0 000 000 011   ; a21: trap 3 (put)
000 001 0 000 001 010   ; a22: load '\n' (new line) (imm. mode)
110101 0 000 000 011   ; a23: trap 3 (put)
110101 0 000 000 001   ; a24: trap 1 (halt)

```

## Addition (immediate mode) – Inside the Machine



## Basic Machine Language Programming

- arithmetic and logic on an accumulator-style (von Neumann) architecture
- instruction operands may be specifiable in different modes:
  - 000: immediate, 001: direct, 010: indirect, 011: indexed, 100: stack
- each program needs a halt instruction at the end to make it terminate (trap 1)
- e.g. addition (immediate mode):

	instruction	AC	mem[a35]
■	load 5	5	?
■	add 14	19	?
	store mem[a35]	19	19
	halt		

- machine code (need to include AT and START to make it valid)

```

000 001 0 000 000 101 ; load 5
000 011 0 000 001 110 ; add 14
001 010 0 000 011 101 ; store mem[a35]
110101 0 000 000 001 ; halt (trap 1)

```

## Addition (direct mode)

- $mem[a3] \leftarrow mem[a1] + mem[a2]$

	instruction	AC	mem[a3]
	load mem[a1]	4	?
(say mem[a1] = 4, mem[a2] = 5)	add mem[a2]	9	?
	store mem[a3]	9	9

- machine code (need to include AT and START)

```

001 001 0 000 000 001 ; load mem[a1]
001 011 0 000 000 010 ; add mem[a2]
001 010 0 000 000 011 ; store mem[a3]
110101 0 000 000 001 ; halt (trap 1)

```



