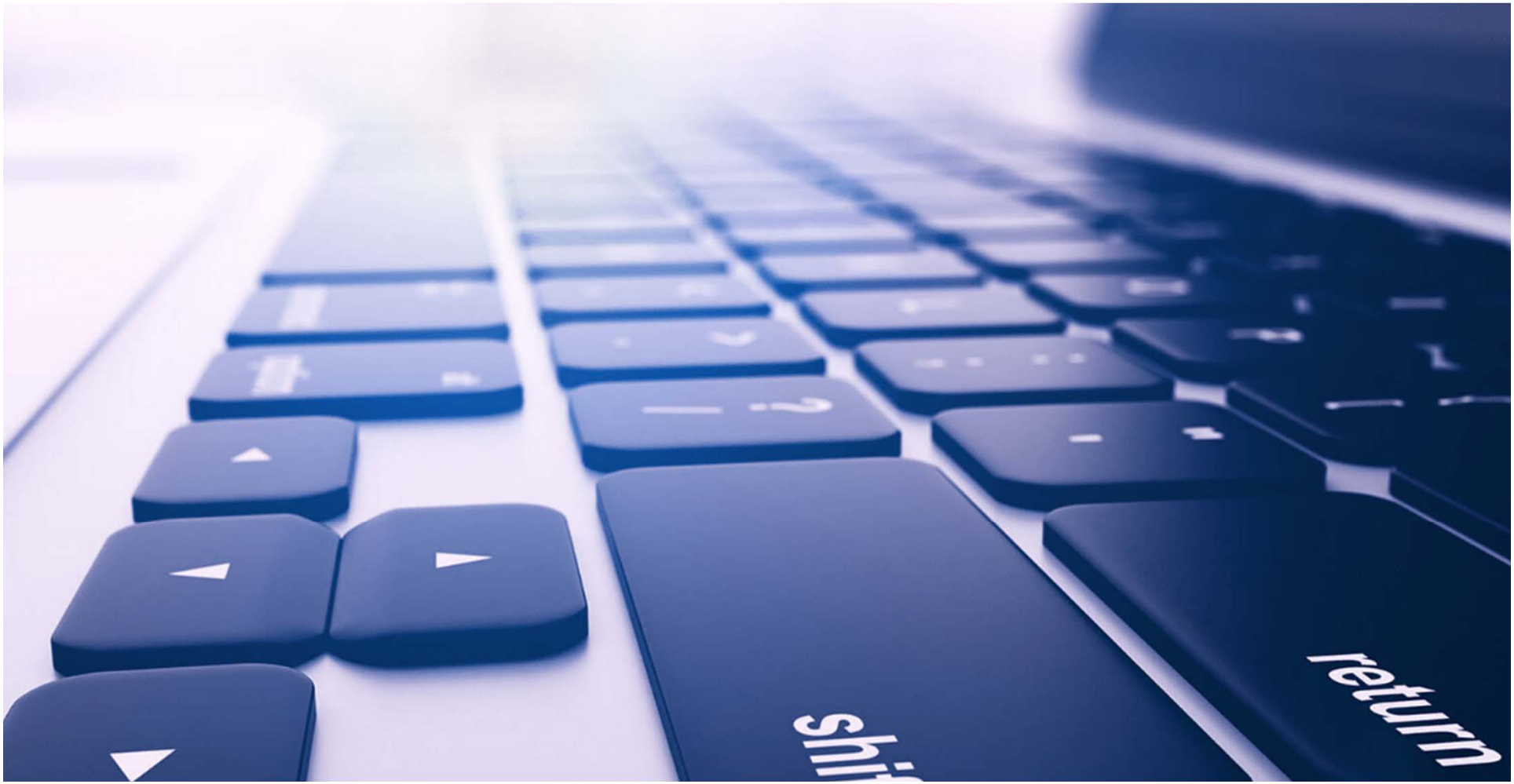

معماری کامپیوتر

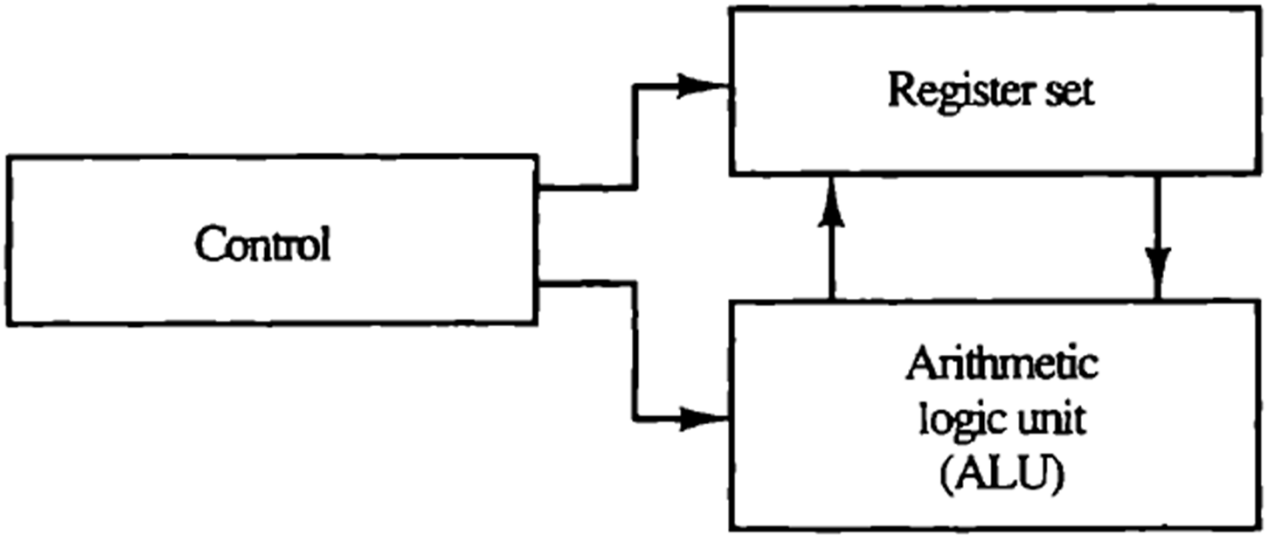
جلسه دهم: طراحی پردازنده

فصل هشتم کتاب موریس مانو – واحـــد پردازش مرکزی

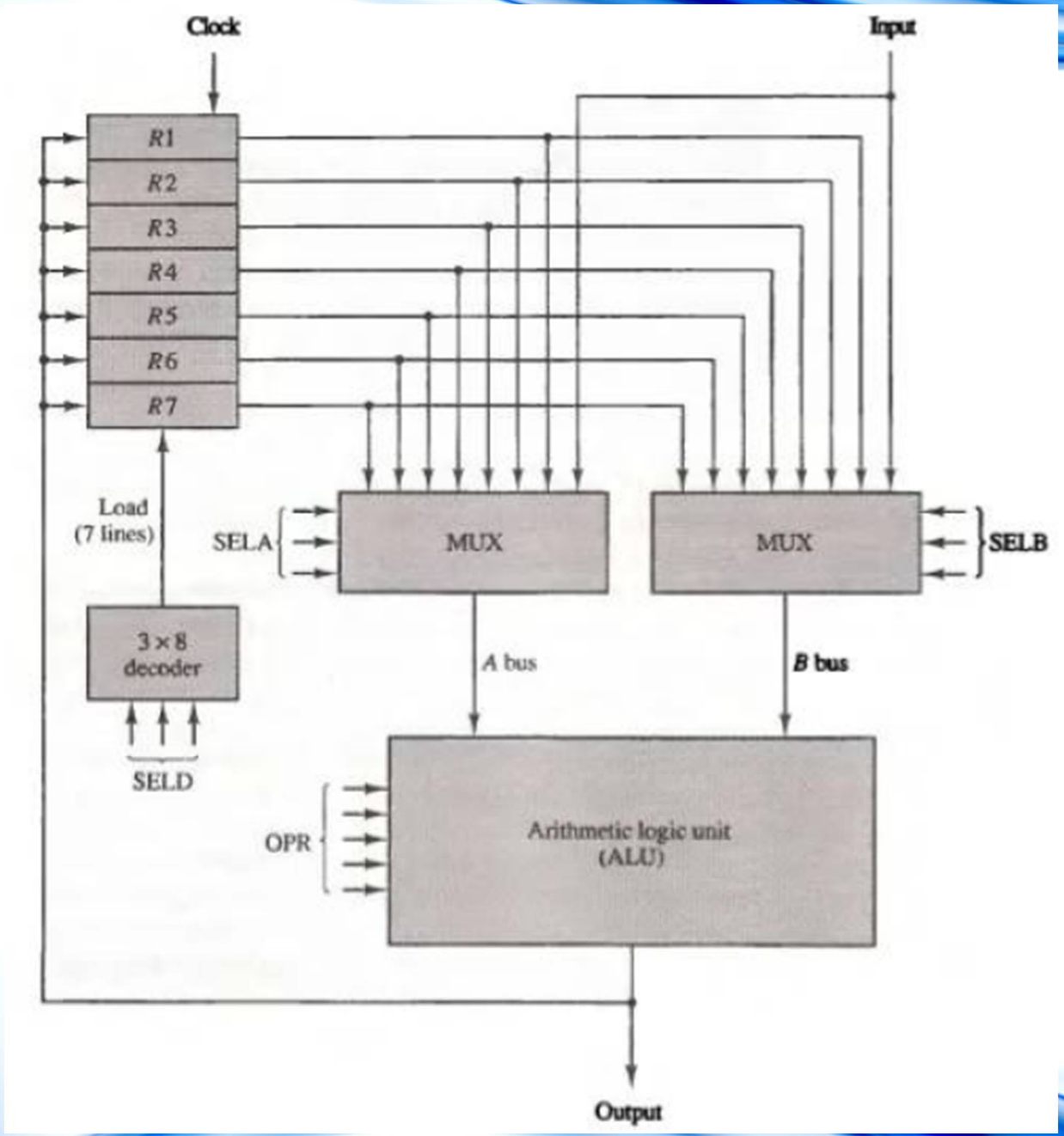


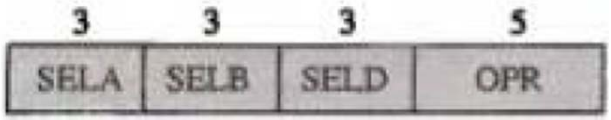
- اجزای اصلی پردازنده
- ساختار عمومی ثباتها
- جداول انتخاب ثبات ها و عملکرد
- ساختار پشته
  - عملیات PUSH
  - عملیات POP
- روش نمایش معکوس لهستانی
- انواع دستورالعملها
  - سه آدرسی
  - دو آدرسی
  - یک آدرسی
  - بدون آدرس
- انواع معماری ها
  - RISC
  - CISC
  - هاروارد
  - فون نویمان

اجزای اصلی پردازنده



ساختار عمومی ثباتها





کـــلمه کنترل

جـــدول انتخاب ثباتها

Encoding of Register Selection Fields

Binary Code	SELA	SELB	SELD
000	Input	Input	None
001	R1	R1	R1
010	R2	R2	R2
011	R3	R3	R3
100	R4	R4	R4
101	R5	R5	R5
110	R6	R6	R6
111	R7	R7	R7

جـــدول عـــملکـــرد

Encoding of ALU Operations

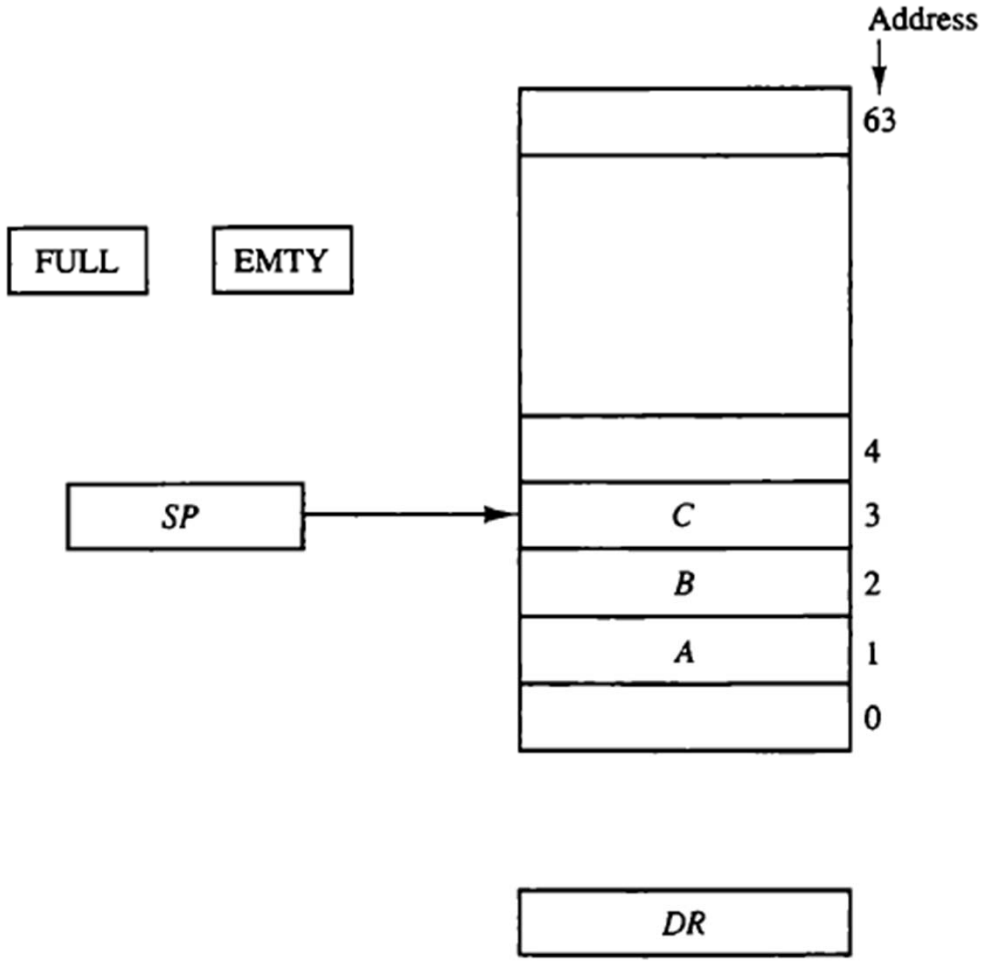
OPR Select	Operation	Symbol
00000	Transfer *A*	TSFA
00001	Increment *A*	INCA
00010	Add *A* + *B*	ADD
00101	Subtract *A* − *B*	SUB
00110	Decrement *A*	DECA
01000	AND *A* and *B*	AND
01010	OR *A* and *B*	OR
01100	XOR *A* and *B*	XOR
01110	Complement *A*	COMA
10000	Shift right *A*	SHRA
11000	Shift left *A*	SHLA

مثالهایی از ریز عملیاتها در پردازنده

Examples of Microoperations for the CPU

Microoperation	Symbolic Designation				Control Word
	SELA	SELB	SELD	OPR	
$R1 \leftarrow R2 - R3$	R2	R3	R1	SUB	010 011 001 00101
$R4 \leftarrow R4 \vee R5$	R4	R5	R4	OR	100 101 100 01010
$R6 \leftarrow R6 + 1$	R6	—	R6	INCA	110 000 110 00001
$R7 \leftarrow R1$	R1	—	R7	TSFA	001 000 111 00000
Output $\leftarrow R2$	R2	—	None	TSFA	010 000 000 00000
Output $\leftarrow$ Input	Input	—	None	TSFA	000 000 000 00000
$R4 \leftarrow$ shl $R4$	R4	—	R4	SHLA	100 000 100 11000
$R5 \leftarrow 0$	R5	R5	R5	XOR	101 101 101 01100

ساختار پشته



عمليات PUSH

$SP \leftarrow SP + 1$	Increment stack pointer
$M[SP] \leftarrow DR$	Write item on top of the stack
If $(SP = 0)$ then $(FULL \leftarrow 1)$	Check if stack is full
$EMTY \leftarrow 0$	Mark the stack not empty

عمليات POP

$DR \leftarrow M[SP]$	Read item from the top of stack
$SP \leftarrow SP - 1$	Decrement stack pointer
If $(SP = 0)$ then $(EMTY \leftarrow 1)$	Check if stack is empty
$FULL \leftarrow 0$	Mark the stack not full

نمایش معکوس لهستانی

$A + B$	Infix notation
$+AB$	Prefix or Polish notation
$AB+$	Postfix or reverse Polish notation

$A * B + C * D$

$A * B + C * D$

$AB * CD * +$

$(A * B)CD * +$

$(A + B) * [C * (D + E) + F]$

$AB + DE + C * F + *$

دو مثال از نمایش معکوس لهستانی

E*(A+B*C+D)

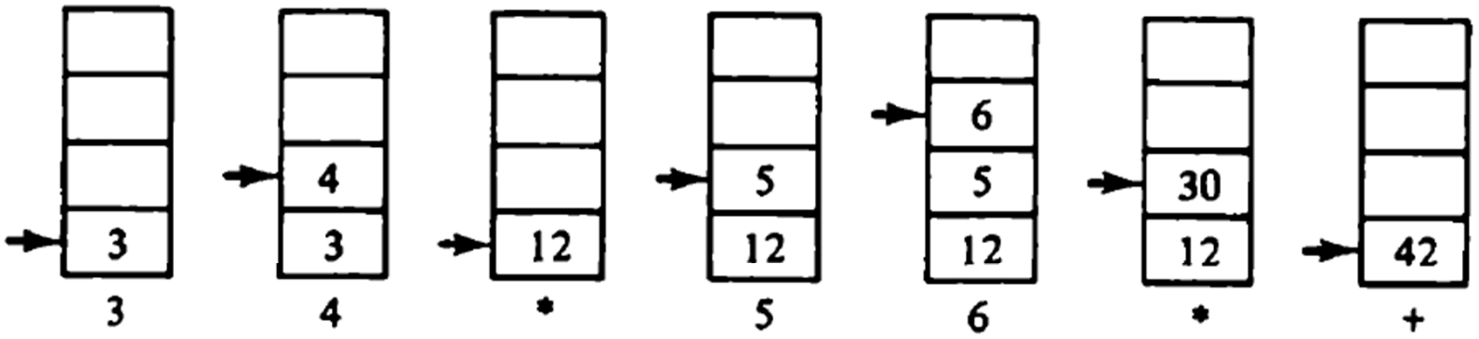
E A B C * + D + *

(A+B*C)+D*E

A B C * + D E * +

کاربرد پشته در محاسبات

Stack operations to evaluate 3 • 4 + 5 • 6.



$(3 * 4) + (5 * 6)$

$34 * 56 * +$

دستورالعمل های سه آدرسی

Three-Address Instructions

```
ADD     R1, A, B       R1 ← M[A] + M[B]
ADD     R2, C, D       R2 ← M[C] + M[D]
MUL     X, R1, R2      M[X] ← R1 * R2
```

دستورالعمل های دو آدرسی

Two-Address Instructions

```
MOV     R1, A       R1 ← M[A]
ADD     R1, B       R1 ← R1 + M[B]
MOV     R2, C       R2 ← M[C]
ADD     R2, D       R2 ← R2 + M[D]
MUL     R1,R2       R1 ← R1 * R2
MOV     X, R1       M[X] ← R1
```

دستورالعمل های یک آدرسی

One-Address Instructions

LOAD	A	AC ← M[A]
ADD	B	AC ← AC + M[B]
STORE	T	M[T] ← AC
LOAD	C	AC ← M[C]
ADD	D	AC ← AC + M[D]
MUL	T	AC ← AC * M[T]
STORE	X	M[X] ← AC

دستورالعملهای بدون آدرس (مبتنی بر پشته)

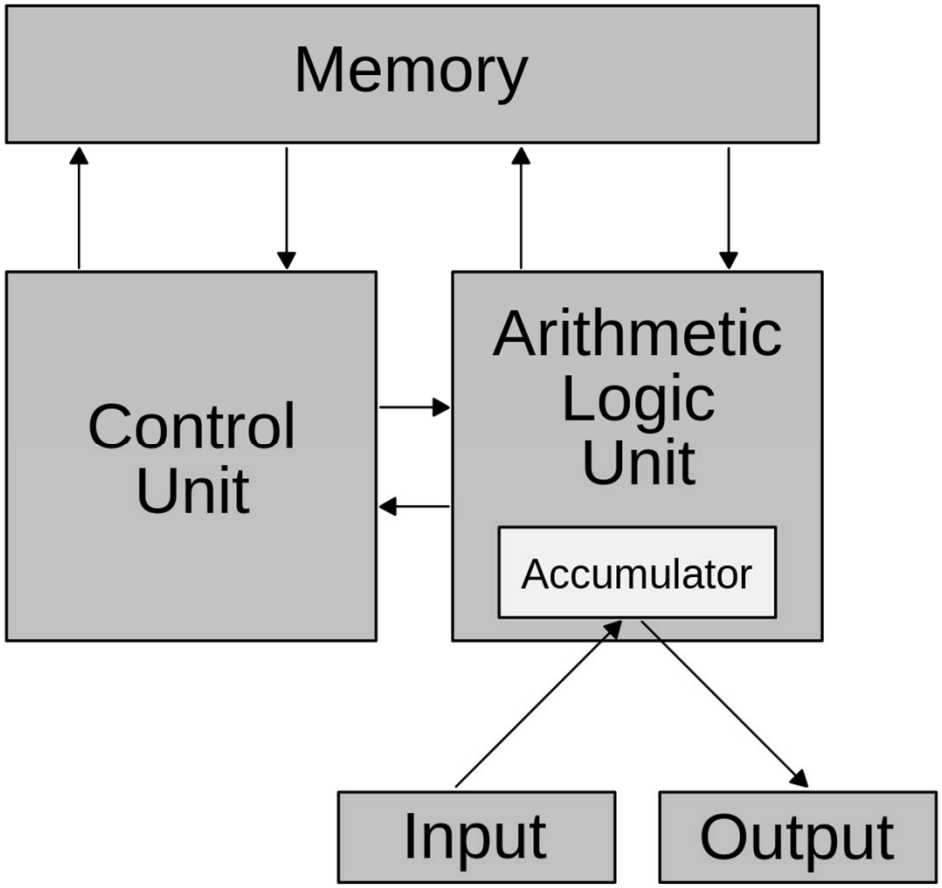
Zero-Address Instructions

```
PUSH    A     TOS ← A
PUSH    B     TOS ← B
ADD           TOS ← (A + B)
PUSH    C     TOS ← C
PUSH    D     TOS ← D
ADD           TOS ← (C + D)
MUL           TOS ← (C + D) * (A + B)
POP     X     M[X] ← TOS
```

مقایسه CISC , RISC

CISC	RISC
Complex Instructions	Simpler Instructions broken down
Emphasis on **hardware**	Emphasis on **software**
Includes multi-clock complex instructions	Single-clock, reduced instruction only
Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	Register to register: "LOAD" and "STORE" are independent instructions
Small code sizes, high cycles per second	**Low cycles** per second, large code sizes
Transistors used for storing complex instructions	Spends more transistors on memory registers

معماری فون نویمان



معماری هاروارد