

# Calculatoare Numerice

– Cursul 5 –

## Calculatorul SAP-1

Facultatea de Automatică și Calculatoare  
Universitatea Politehnica București

# Unitatea de Control

---

- Unitatea de control este o componentă cheie a calculatorului și este responsabilă cu generarea cuvintelor de control ce citesc, decodifică și execută fiecare instrucțiune.
- Pe parcursul execuției unei instrucțiuni, procesorul trece prin mai multe stări (timing states), intervale de timp în care conținutul registrelor se modifică
- Ring Counter are o ieșire de forma:  $T = T_6T_5T_4T_3T_2T_1$
- La începutul rulării calculatorului, cuvântul T are valoarea  $000001 = T_1$
- Impulsuri succesive ale ceasului produc stări succesive ale Ring Counter.
- Este important de știut că Ring Counter se incrementează pe **frontul negativ** al ceasului. De aceea, un front pozitiv al ceasului apare **întotdeauna** în mijlocul unei stări T.

# Address State ( $T = 000001 = T_1$ )

- $T_1$  este denumită și **Starea Adresă**, deoarece adresa din Program Counter este transferată în Memory Address Register în această stare.
- În această stare, semnalele  $E_P$  și  $L_M$  sunt active, toți ceilalți biți de control fiind inactivi. Acest lucru înseamnă că unitatea de control trimite un **cuvânt de control** cu valoarea **5E3H** în această stare.

$$\begin{aligned} \text{CON} &= C_P \ E_P \ \overline{L_M} \ \overline{C_E} \\ &= 0 \ 1 \ 0 \ 1 \\ &= 5 \end{aligned}$$

$$\begin{aligned} &\overline{L_I} \ \overline{E_I} \ \overline{L_A} \ E_A \\ &1 \ 1 \ 1 \ 0 \\ &E \end{aligned}$$

$$\begin{aligned} &S_U \ L_U \ \overline{E_B} \ \overline{L_O} \\ &0 \ 0 \ 1 \ 1 \\ &3 \end{aligned}$$

# Increment State ( $T = 000010 = T_2$ )

- $T_2$  este denumită și **Starea de Incrementare**, pentru că în această stare Program Counter este incrementat
- Doar bitul  $C_P$  este activ în această stare, deci unitatea de control va produce cuvântul **BE3H**

$$\begin{array}{l} \text{CON} = \\ = \\ = \end{array} \begin{array}{cccc} C_P & E_P & \overline{L_M} & \overline{C_E} \\ 1 & 0 & 1 & 1 \\ & & & \text{B} \end{array} \quad \begin{array}{cccc} \overline{L_I} & \overline{E_I} & \overline{L_A} & E_A \\ 1 & 1 & 1 & 0 \\ & & & \text{E} \end{array} \quad \begin{array}{cccc} S_U & L_U & \overline{E_B} & \overline{L_O} \\ 0 & 0 & 1 & 1 \\ & & & \text{3} \end{array}$$

# Memory State ( $T = 000100 = T_3$ )

- $T_3$  poartă numele de **Starea Memorie**, pentru că instrucțiunea adresată din RAM este tranferată în registrul instrucțiune.
- Singurii biți de control activi în această stare sunt  $C_E$  și  $L_I$ , deci cuvântul de control al UC este **263H**

$$\begin{aligned} \text{CON} &= C_P \ E_P \ \overline{L_M} \ \overline{C_E} \\ &= 0 \ 0 \ 1 \ 0 \\ &= \quad \quad 2 \end{aligned}$$

$$\begin{aligned} &\overline{L_I} \ \overline{E_I} \ \overline{L_A} \ E_A \\ &0 \ 1 \ 1 \ 0 \\ &\quad \quad 6 \end{aligned}$$

$$\begin{aligned} &S_U \ L_U \ \overline{E_B} \ \overline{L_O} \\ &0 \ 0 \ 1 \ 1 \\ &\quad \quad 3 \end{aligned}$$

# Fetch Cycle

---

- Stările **adresă**, **incrementare** și **memorie** ale SAP-1 compun un **ciclu de fetch**
- În starea adresă,  $E_P$  și  $L_M$  sunt active, deci Program Counter-ul setează MAR prin magistrala W
- Un front pozitiv al ceasului se petrece în mijlocul stării adresă, ce produce încărcarea MAR cu conținutul PC
- În **starea de incrementare**,  $C_P$  este singurul bit activ, acest lucru setează PC să numere la fiecare front pozitiv de ceas. La mijlocul stării de incrementare, un front pozitiv al ceasului va avansa PC cu 1
- În **starea memorie**,  $C_E$  și  $L_I$  sunt active. Cuvântul RAM adresat setează registrul instrucțiune prin magistrala W. La mijlocul stării memorie, un front pozitiv al ceasului încarcă registrul instrucțiune cu valoarea cuvântului adresat din RAM.

# Execution Cycle

---

- Următoarele trei stări ( $T_4$ ,  $T_5$  și  $T_6$ ) formează ciclul de execuție al SAP-1
- Transferurile între registre în acest ciclu depind de tipul instrucțiunii executate
- De exemplu, **LDA 9H** necesită transferuri din registre diferite decât **ADD BH**
- Urmează rutine de control pentru fiecare din instrucțiunile SAP-1

# Micro și Macro-Instrucțiuni

---

- Unitatea de control generează cuvinte de control în timpul fiecărei din cele șase stări T.
- Pentru că fiecare cuvânt de control generează un mic pas în procesare, acesta poartă numele de **micro-instrucțiune**
- Instrucțiunile cu care am proiectat calculatorul (LDA, ADD, SUB etc.) sunt uneori denumite și **macro-instrucțiuni**, pentru a le distinge.
- Fiecare **macro-instrucțiune** SAP-1 este formată din cel puțin trei **micro-instrucțiuni**.

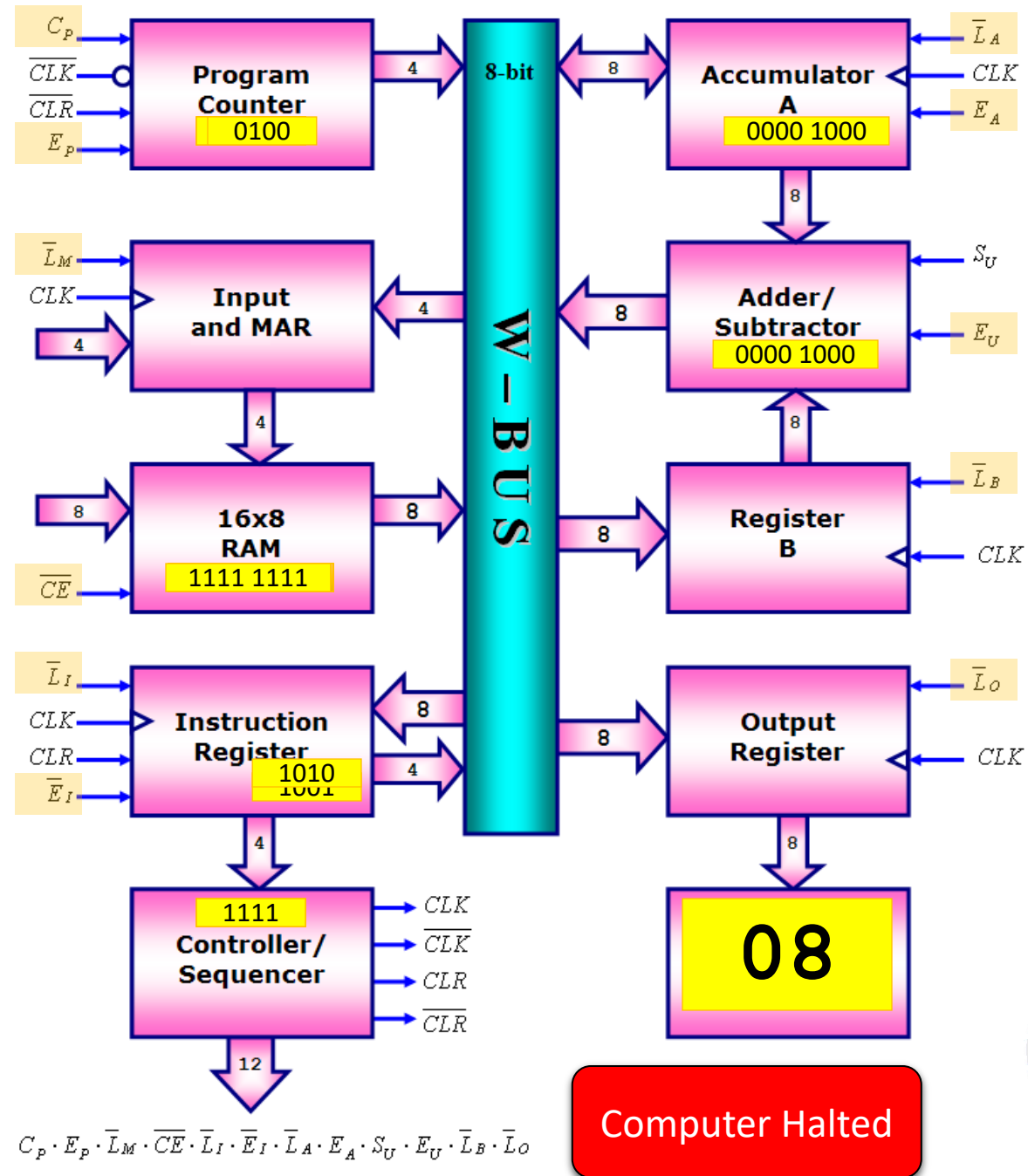


Fetch and Execute Cycle of SAP-1				
Macro Inst.	T State	Micro Operation	Active	CON
All Instructions	T <sub>1</sub>	MAR ← PC	L' <sub>M</sub> , E <sub>P</sub>	5E3H
	T <sub>2</sub>	PC ← PC+1	C <sub>P</sub>	BE3H
	T <sub>3</sub>	IR ← RAM[MAR]	CE', L' <sub>I</sub>	263H
LDA	T <sub>4</sub>	MAR ← IR(3...0)	L' <sub>M</sub> , E' <sub>I</sub>	1A3H
	T <sub>5</sub>	ACC ← RAM[MAR]	CE', L' <sub>A</sub>	2C3H
	T <sub>6</sub>	None	None	3E3H
ADD	T <sub>4</sub>	MAR ← IR(3...0)	L' <sub>M</sub> , E' <sub>I</sub>	1A3H
	T <sub>5</sub>	B ← RAM[MAR]	CE', L' <sub>B</sub>	2E1H
	T <sub>6</sub>	ACC ← ACC+B	L' <sub>A</sub> , E <sub>U</sub>	3C7H
SUB	T <sub>4</sub>	MAR ← IR(3...0)	L' <sub>M</sub> , E' <sub>I</sub>	1A3H
	T <sub>5</sub>	B ← RAM[MAR]	CE', L' <sub>B</sub>	2E1H
	T <sub>6</sub>	ACC ← ACC - B	L' <sub>A</sub> , S <sub>U</sub> , E <sub>U</sub>	3CFH
OUT	T <sub>4</sub>	OUT ← ACC	E <sub>A</sub> , L' <sub>O</sub>	3F2H
	T <sub>5</sub>	None	None	3E3H
	T <sub>6</sub>	None	None	3E3H
HLT	T <sub>3</sub>	None	HLT'	263H

# SAP-1

Simularea execuției

➔ LDA 9H  
ADD AH  
OUT  
HLT



Computer Halted

T<sub>3</sub>

# Acknowledgements

---

- Aceste slide-uri conțin materiale aparținând:
  - Arvind (MIT)
  - Krste Asanovic (MIT/UCB)
  - Joel Emer (Intel/MIT)
  - James Hoe (CMU)
  - John Kubiatowicz (UCB)
  - David Patterson (UCB)
  - Behrooz Parhami (UCSB)
  - Rashid Chishti (IIUI)
- MIT material derived from course 6.823
- UCB material derived from course CS252