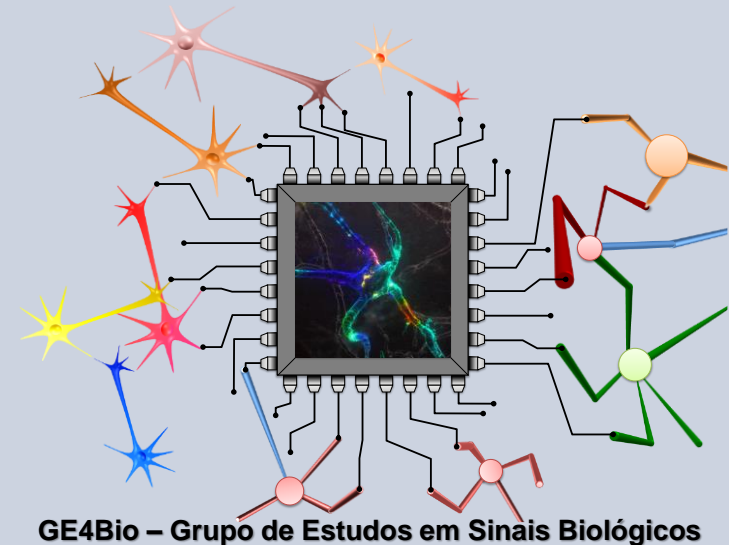


Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Sistemas de Computação

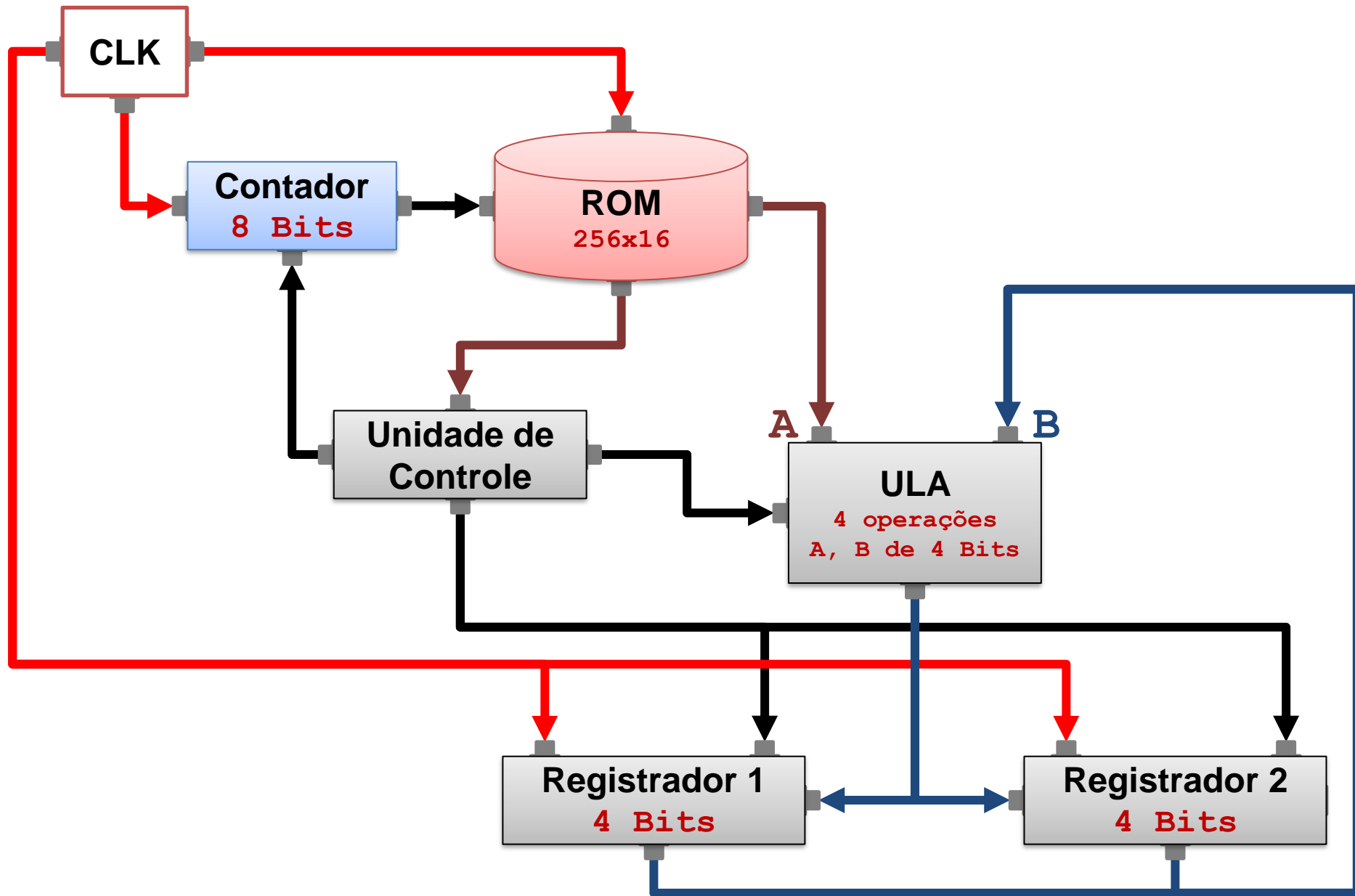
SSC108
Prática em Sistemas Digitais

Projeto CPU - 02



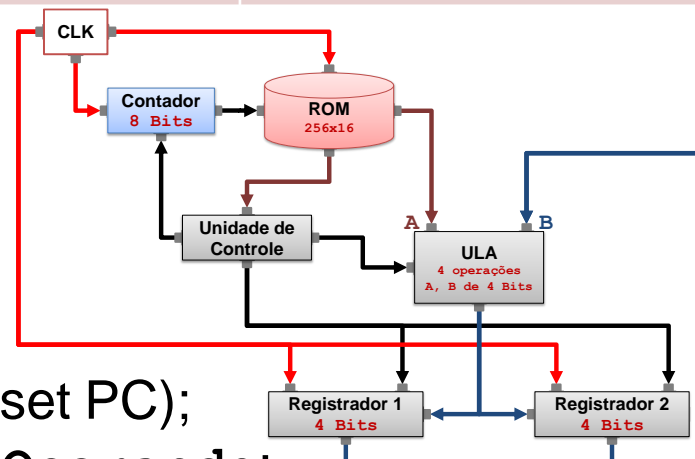
Prof.Dr. Danilo Spatti

São Carlos



15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RgTO				RgIN			JMP		ULA		Disponível			Operando	

- **RgTO**: Registrador de destino (4 bits)
- **RgIN**: Registrador de origem (4 bits)
- **JMP**: uma das 4 opções abaixo:
 - 00: Operação de ULA;
 - 01: Reinicia Registradores;
 - 10: Reinicia o contador de memória (reset PC);
 - 11: Jump para posição de memória do **Operando**;
- **ULA**: uma das 4 operações abaixo:
 - 00: $RgTO \leftarrow RgIN + Operando$;
 - 01: $RgTO \leftarrow Operando * 2$;
 - 10: $RgTO \leftarrow RgIN - Operando$;
 - 11: $RgTO \leftarrow Operando / 2$;
- **Disponível**: Disponível para melhorar a CPU caso queiram
- **Operando**: 4 bits diretamente da memória na ULA



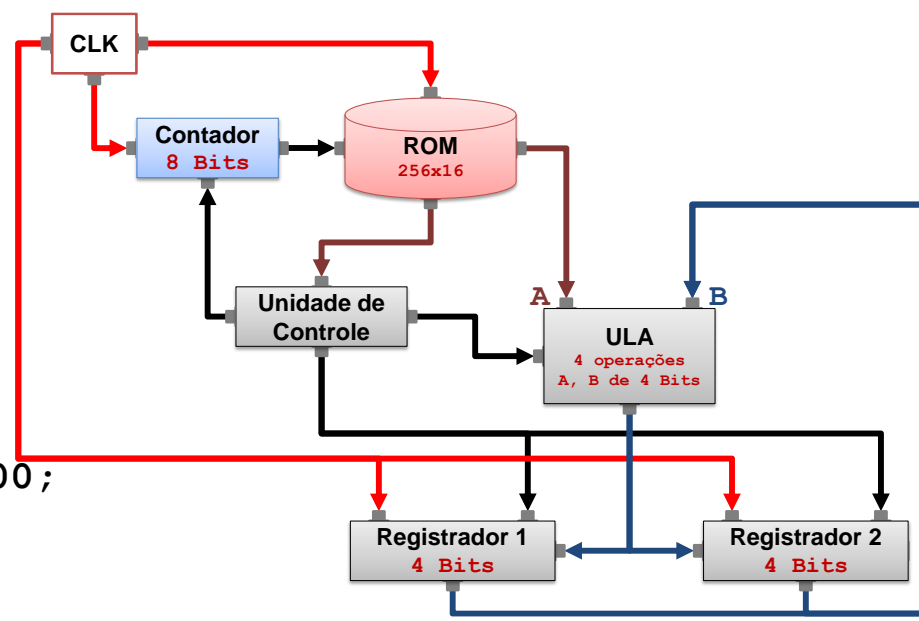
```
WIDTH=16;
DEPTH=256;
```

```
ADDRESS_RADIX=UNS;
DATA_RADIX=BIN;
```

```
CONTENT BEGIN
0      :    01100000000000101;
1      :    10010001000000001;
2      :    01100010000000010;
3      :    01100011000000100;
4      :    0000110000001000;
[5..7] :    0000000000000000;
8      :    0000010000000000;
9      :    01100000000000001;
10     :    10010000000000001;
11     :    10010000000000010;
12     :    01100011000000010;
13     :    01100001000000010;
14     :    10010011000000010;
15     :    01100010000000000;
16     :    01100000000000001;
17     :    00001000000000000;
[18..255] : 0000000000000000;
END;
```

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RgTO				RgIN		JMP		ULA		Disponível				Operando	

- **RgTO**: Registrador de destino (4 bits)
- **RgIN**: Registrador de origem (4 bits)
- **JMP**: uma das 4 opções abaixo:
 - 00: Operação de ULA;
 - 01: Reinicia Registradores;
 - 10: Reinicia o contador de memória (reset PC);
 - 11: Jump para posição de memória do **Operando**;
- **ULA**: uma das 4 operações abaixo:
 - 00: $RgTO \leftarrow RgIN + Operando$;
 - 01: $RgTO \leftarrow Operando * 2$;
 - 10: $RgTO \leftarrow RgIN - Operando$;
 - 11: $RgTO \leftarrow Operando / 2$;
- **Disponível**: Disponível para melhorar a CPU caso queiram
- **Operando**: 4 bits diretamente da memória na ULA



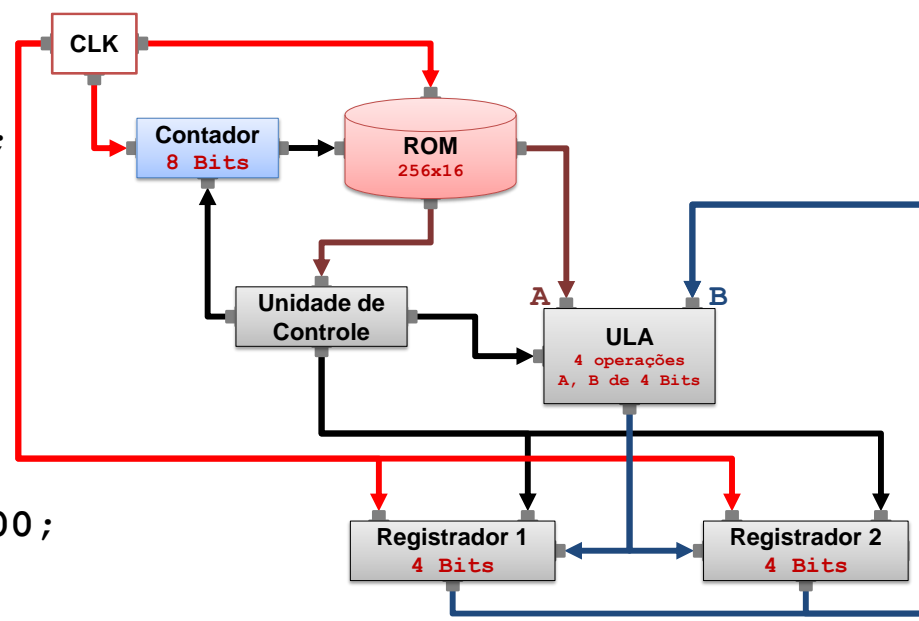
```
WIDTH=16;
DEPTH=256;
```

```
ADDRESS_RADIX=UNS;
DATA_RADIX=BIN;
```

```
CONTENT BEGIN
0      :    01100000000000100;
1      :    10010001000000010;
2      :    01100010000000001;
3      :    01100011000000101;
4      :    10010000000000010;
5      :    01100000000000010;
6      :    10010000000000011;
7      :    00000100000000000;
8      :    00001100000001101;
[9..12] :    00000000000000000;
13     :    10010000000000010;
14     :    01100011000000010;
15     :    01100001000000001;
16     :    10010011000000010;
17     :    01100010000000000;
18     :    01100000000000011;
19     :    00001000000000000;
[20..255] :    00000000000000000;
END;
```

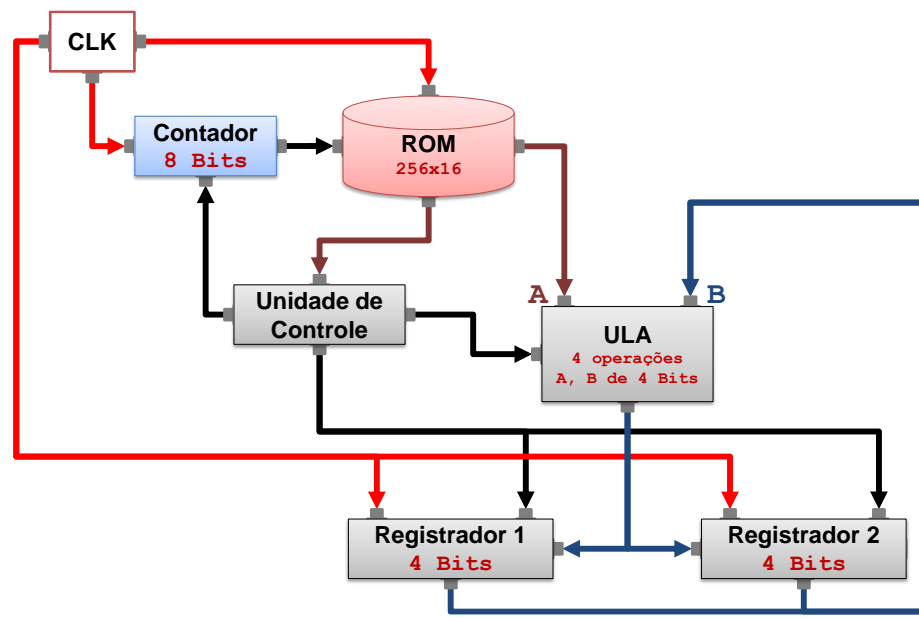
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RgTO				RgIN		JMP		ULA		Disponível				Operando	

- **RgTO**: Registrador de destino (4 bits)
- **RgIN**: Registrador de origem (4 bits)
- **JMP**: uma das 4 opções abaixo:
 - 00: Operação de ULA;
 - 01: Reinicia Registradores;
 - 10: Reinicia o contador de memória (reset PC);
 - 11: Jump para posição de memória do **Operando**;
- **ULA**: uma das 4 operações abaixo:
 - 00: $RgTO \leftarrow RgIN + Operando$;
 - 01: $RgTO \leftarrow Operando * 2$;
 - 10: $RgTO \leftarrow RgIN - Operando$;
 - 11: $RgTO \leftarrow Operando / 2$;
- **Disponível**: Disponível para melhorar a CPU caso queiram
- **Operando**: 4 bits diretamente da memória na ULA



Count	Memória				R1	R2	Instrução
0	6	0	0	5	5	0	$R1 \leftarrow R2 + 5$
1	9	1	0	1	5	2	$R2 \leftarrow 1 \times 2$
2	6	2	0	2	0	2	$R1 \leftarrow R2 - 2$
3	6	3	0	4	2	2	$R1 \leftarrow 4 \div 2$
4	0	c	0	8	2	2	<i>Jump #8</i>
8	0	4	0	0	0	0	<i>CLR R1 e R2</i>
9	6	0	0	1	1	0	$R1 \leftarrow R2 + 1$
10	9	0	0	1	1	2	$R2 \leftarrow R1 + 1$
11	9	0	0	2	1	3	$R2 \leftarrow R1 + 2$
12	6	3	0	2	1	3	$R1 \leftarrow 2 \div 2$
13	6	1	0	2	4	3	$R1 \leftarrow 2 \times 2$
14	9	3	0	2	4	1	$R2 \leftarrow 2 \div 2$
15	6	2	0	0	1	1	$R1 \leftarrow R2 - 0$
16	6	0	0	1	2	1	$R1 \leftarrow R2 + 1$
17	0	8	0	0	2	1	<i>CLR contador</i>

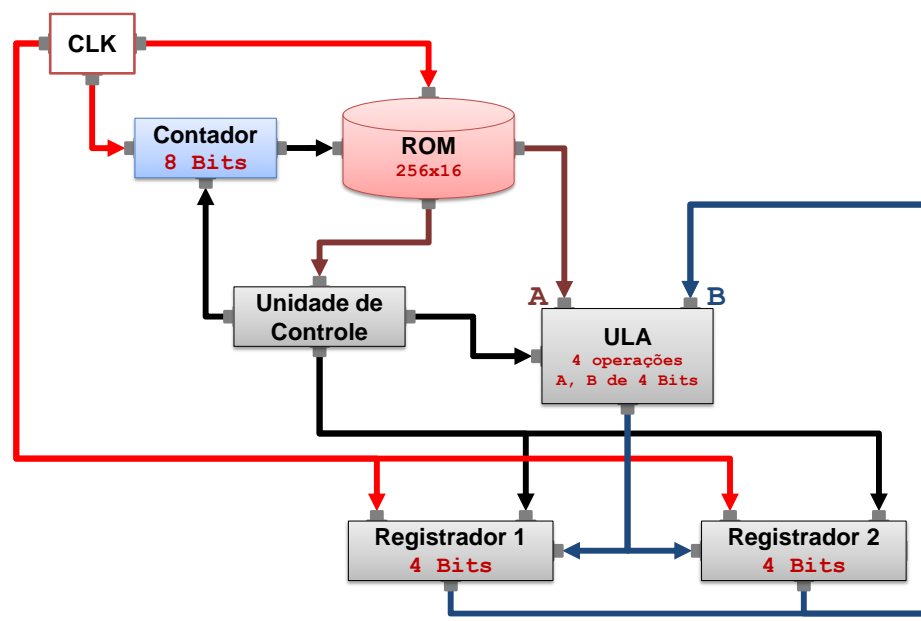
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RgTO				RgIN		JMP		ULA		Disponível			Operando		
<ul style="list-style-type: none">• RgTO: Registrador de destino (4 bits)• RgIN: Registrador de origem (4 bits)• JMP: uma das 4 opções abaixo:<ul style="list-style-type: none">• 00: Operação de ULA;• 01: Reinicia Registradores;• 10: Reinicia o contador de memória (reset PC);• 11: Jump para posição de memória do Operando;• ULA: uma das 4 operações abaixo:<ul style="list-style-type: none">• 00: $RgTO \leftarrow RgIN + \text{Operando}$;• 01: $RgTO \leftarrow \text{Operando} * 2$;• 10: $RgTO \leftarrow RgIN - \text{Operando}$;• 11: $RgTO \leftarrow \text{Operando} / 2$;• Disponível: Disponível para melhorar a CPU caso queiram• Operando: 4 bits diretamente da memória na ULA															

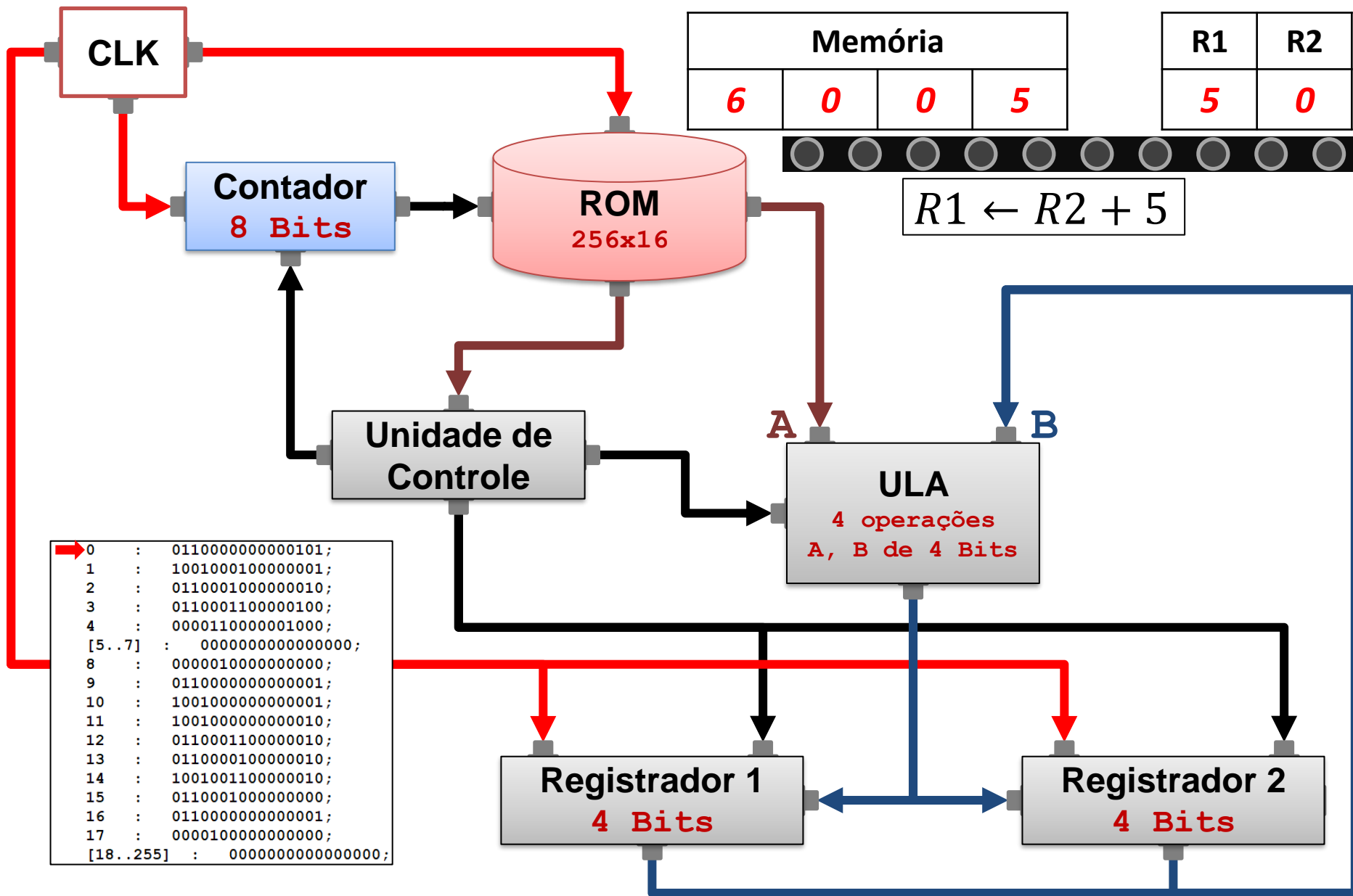


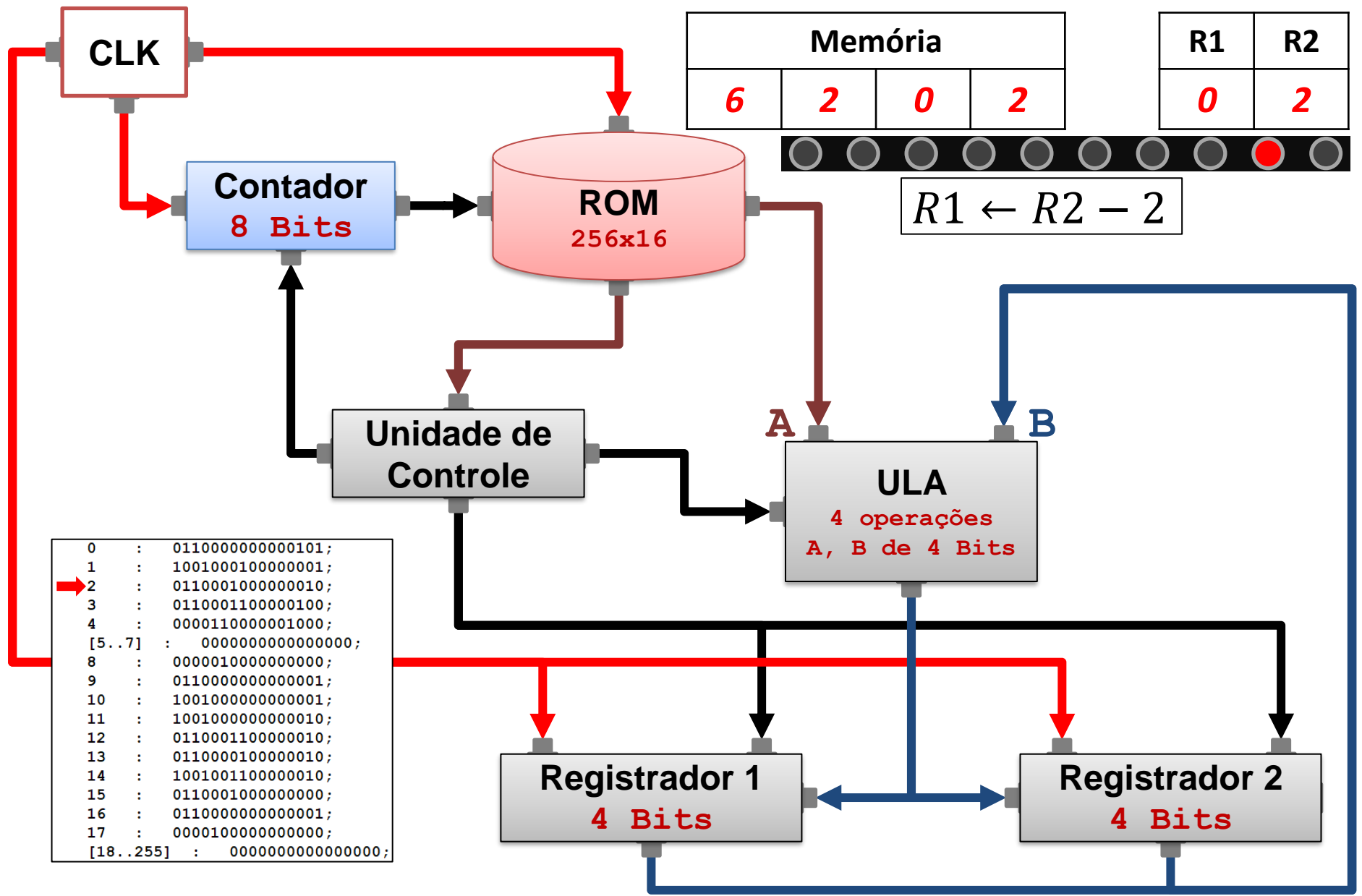
Count	Memória				R1	R2	Instrução
0	6	0	0	4	4	0	$R1 \leftarrow R2 + 4$
1	9	1	0	2	4	4	$R2 \leftarrow 2 \times 2$
2	6	2	0	1	3	4	$R1 \leftarrow R2 - 1$
3	6	3	0	5	2	4	$R1 \leftarrow 5 \div 2$
4	9	0	0	2	2	4	$R2 \leftarrow R1 + 2$
5	6	0	0	2	6	4	$R1 \leftarrow R2 + 2$
6	9	0	0	3	6	9	$R2 \leftarrow R1 + 3$
7	0	4	0	0	0	0	CLR R1 e R2
8	0	c	0	d	0	0	Jump #d
13	9	0	0	2	0	2	$R2 \leftarrow R1 + 2$
14	6	3	0	2	1	2	$R1 \leftarrow 2 \div 2$
15	6	1	0	1	2	2	$R1 \leftarrow 1 \times 2$
16	9	3	0	2	2	1	$R2 \leftarrow 2 \div 2$
17	6	2	0	0	1	1	$R1 \leftarrow R2 - 0$
18	6	0	0	3	4	1	$R1 \leftarrow R2 + 3$
19	0	8	0	0	4	1	CLR contador

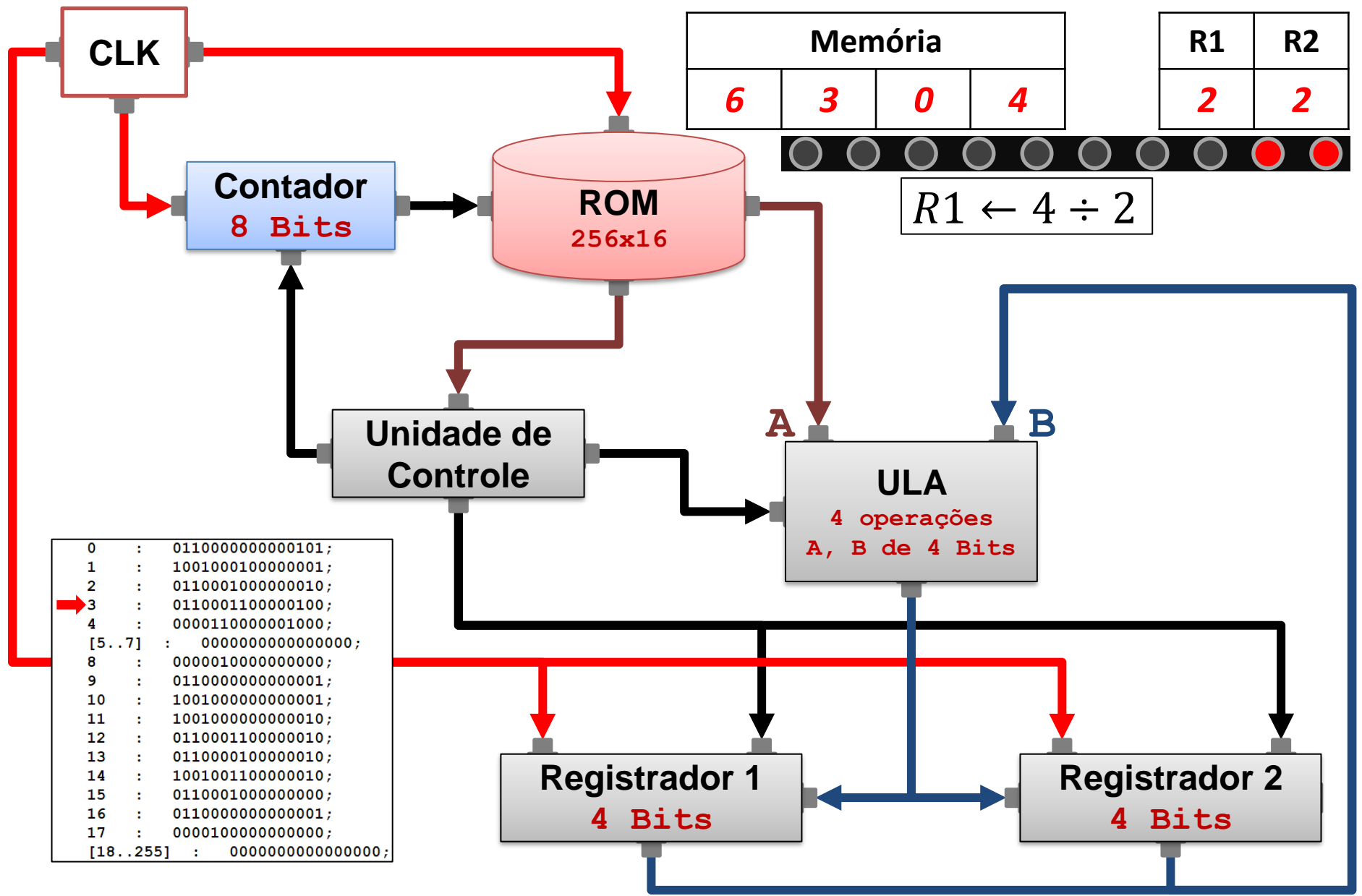
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RgTO				RgIN		JMP		ULA		Disponível			Operando		

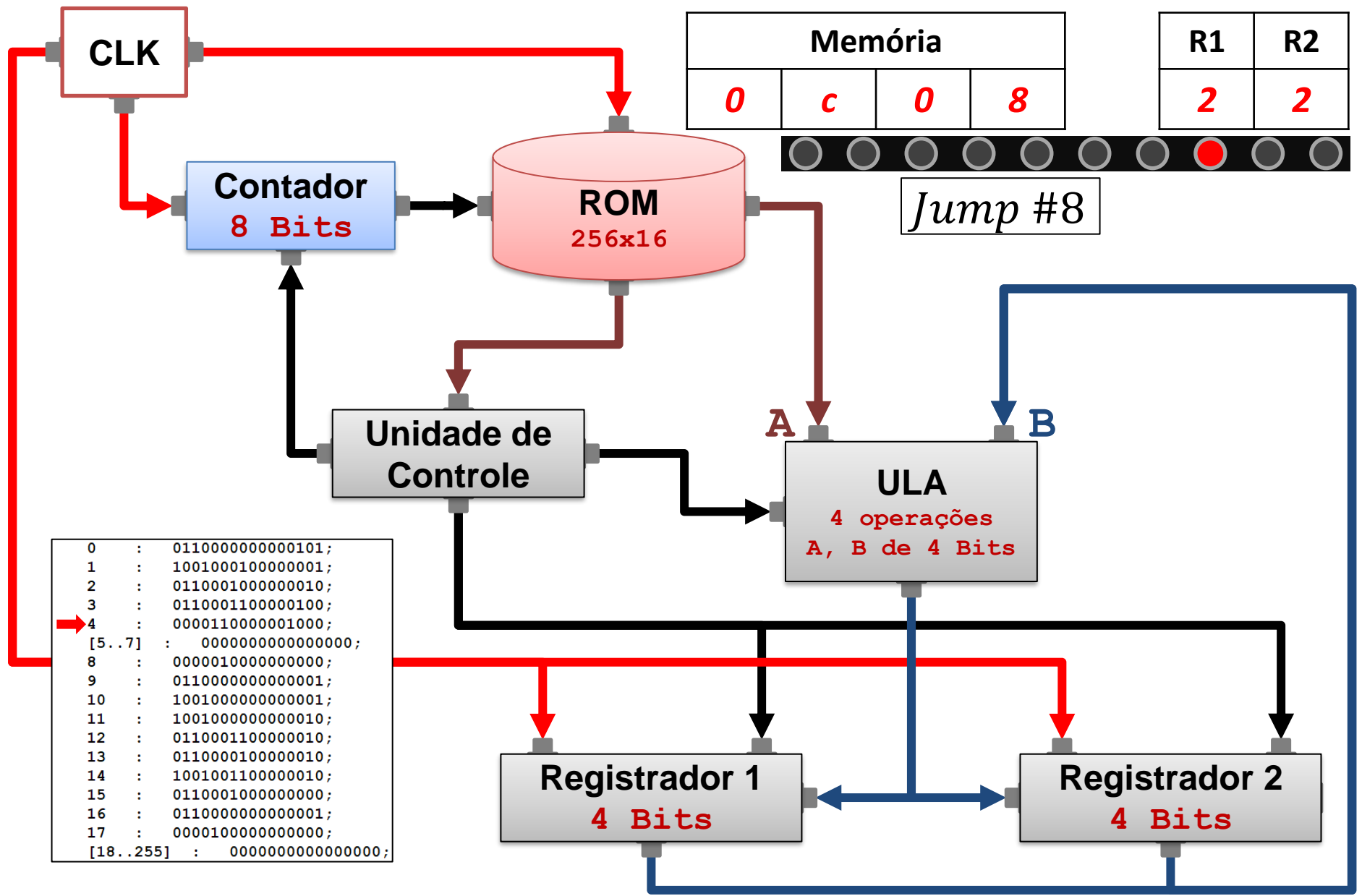
- **RgTO**: Registrador de destino (4 bits)
- **RgIN**: Registrador de origem (4 bits)
- **JMP**: uma das 4 opções abaixo:
 - 00: Operação de ULA;
 - 01: Reinicia Registradores;
 - 10: Reinicia o contador de memória (reset PC);
 - 11: Jump para posição de memória do **Operando**;
- **ULA**: uma das 4 operações abaixo:
 - 00: $RgTO \leftarrow RgIN + \text{Operando}$;
 - 01: $RgTO \leftarrow \text{Operando} * 2$;
 - 10: $RgTO \leftarrow RgIN - \text{Operando}$;
 - 11: $RgTO \leftarrow \text{Operando} / 2$;
- **Disponível**: Disponível para melhorar a CPU caso queiram
- **Operando**: 4 bits diretamente da memória na ULA

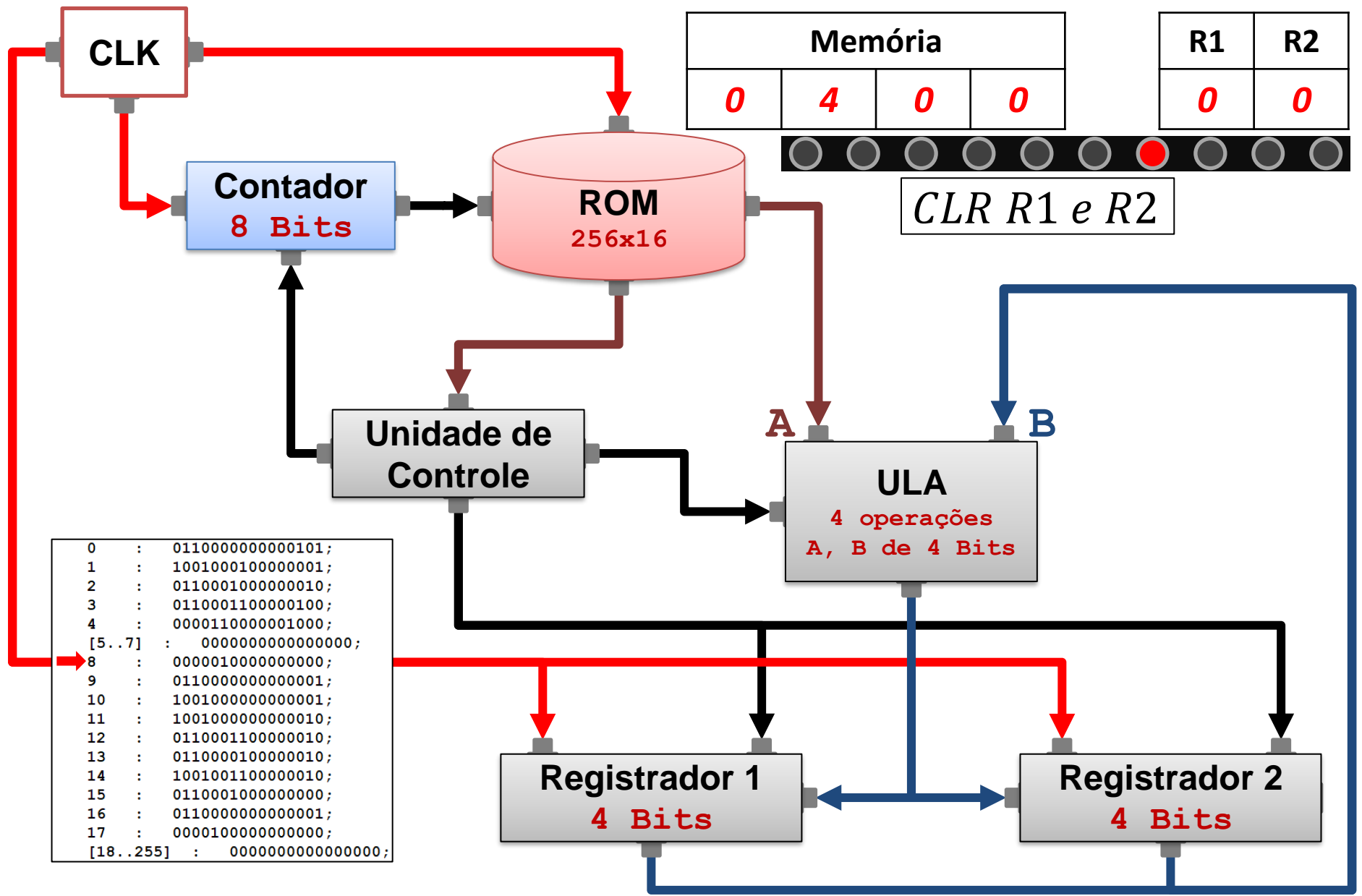


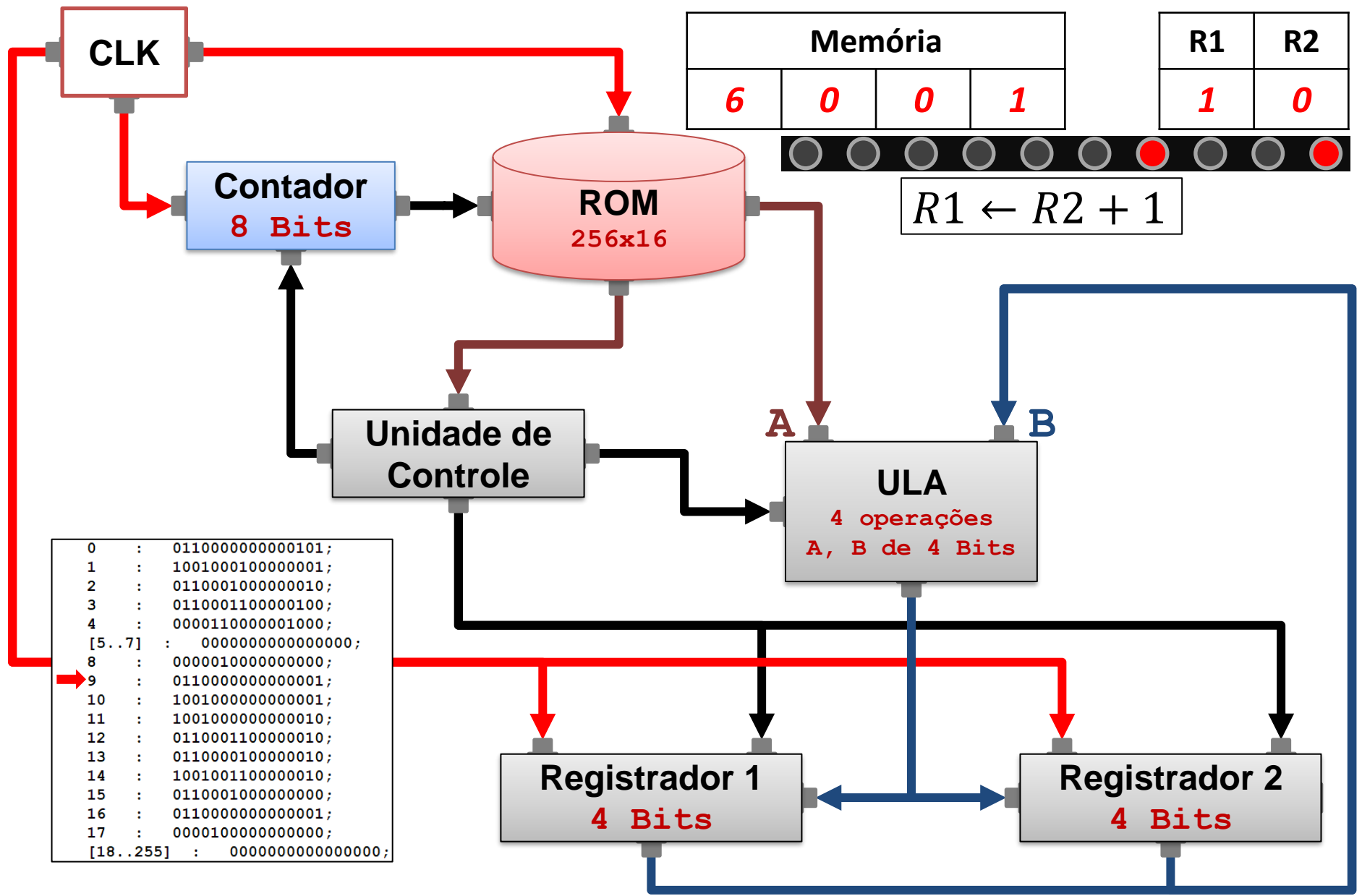


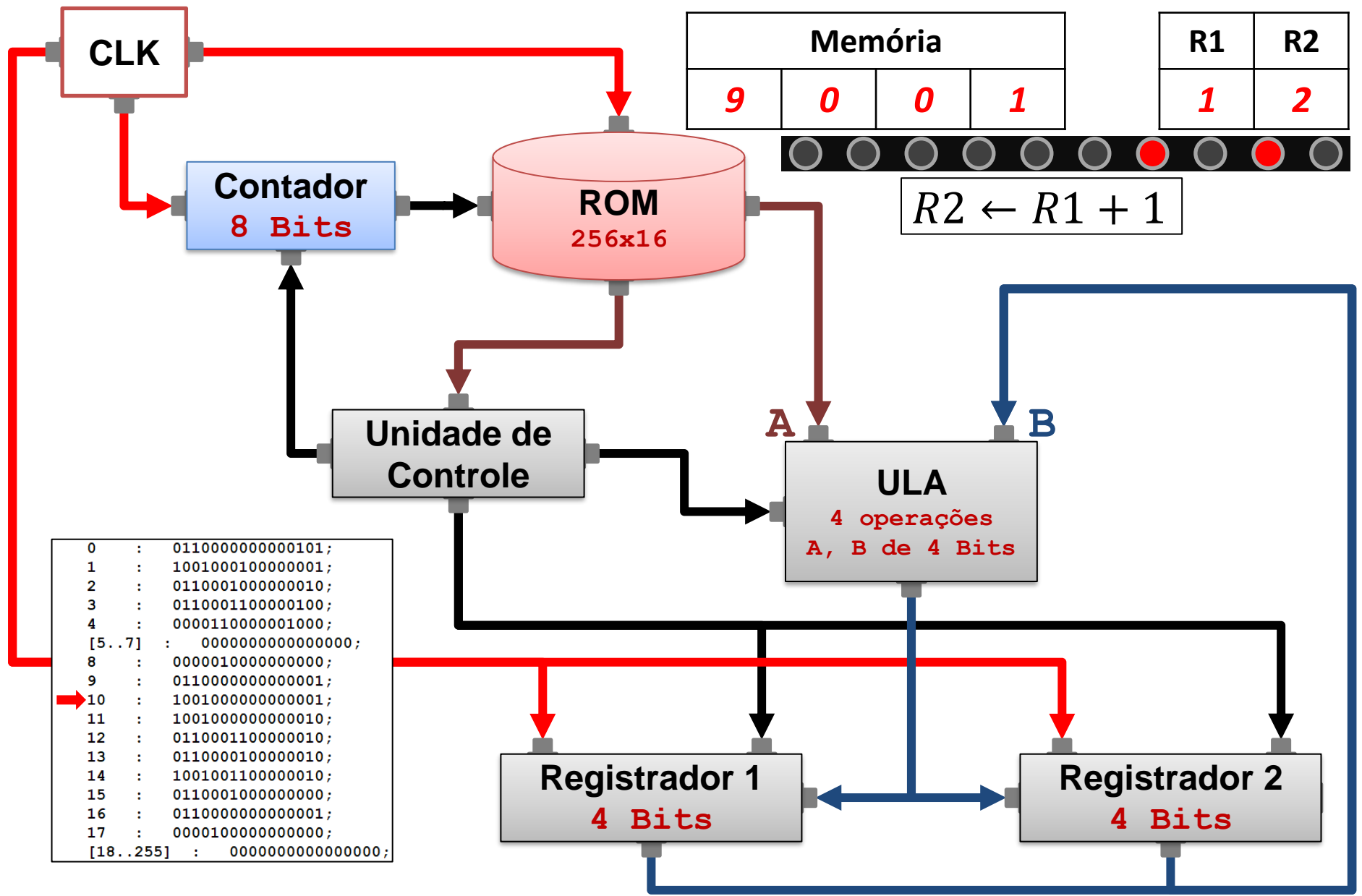


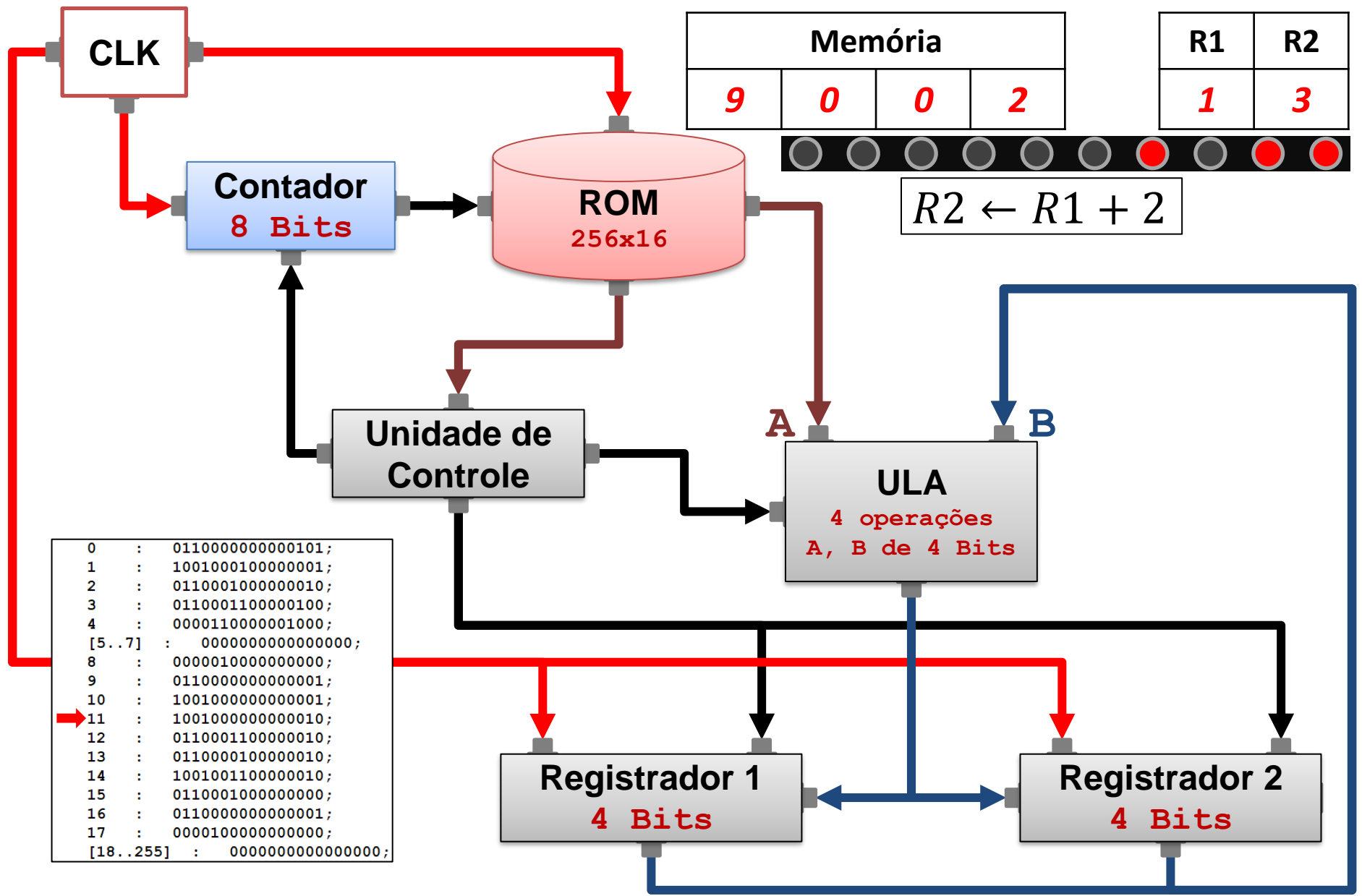


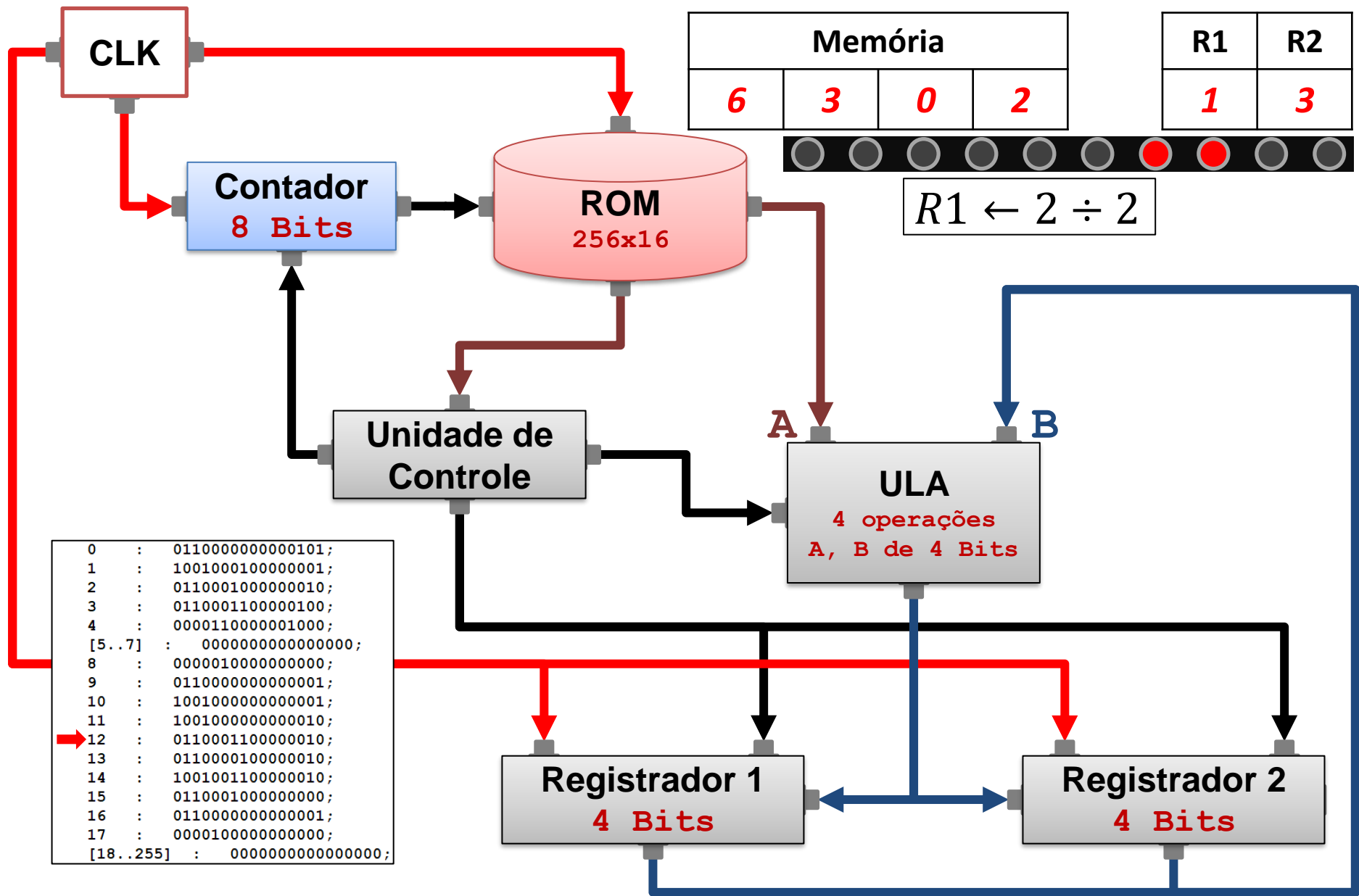


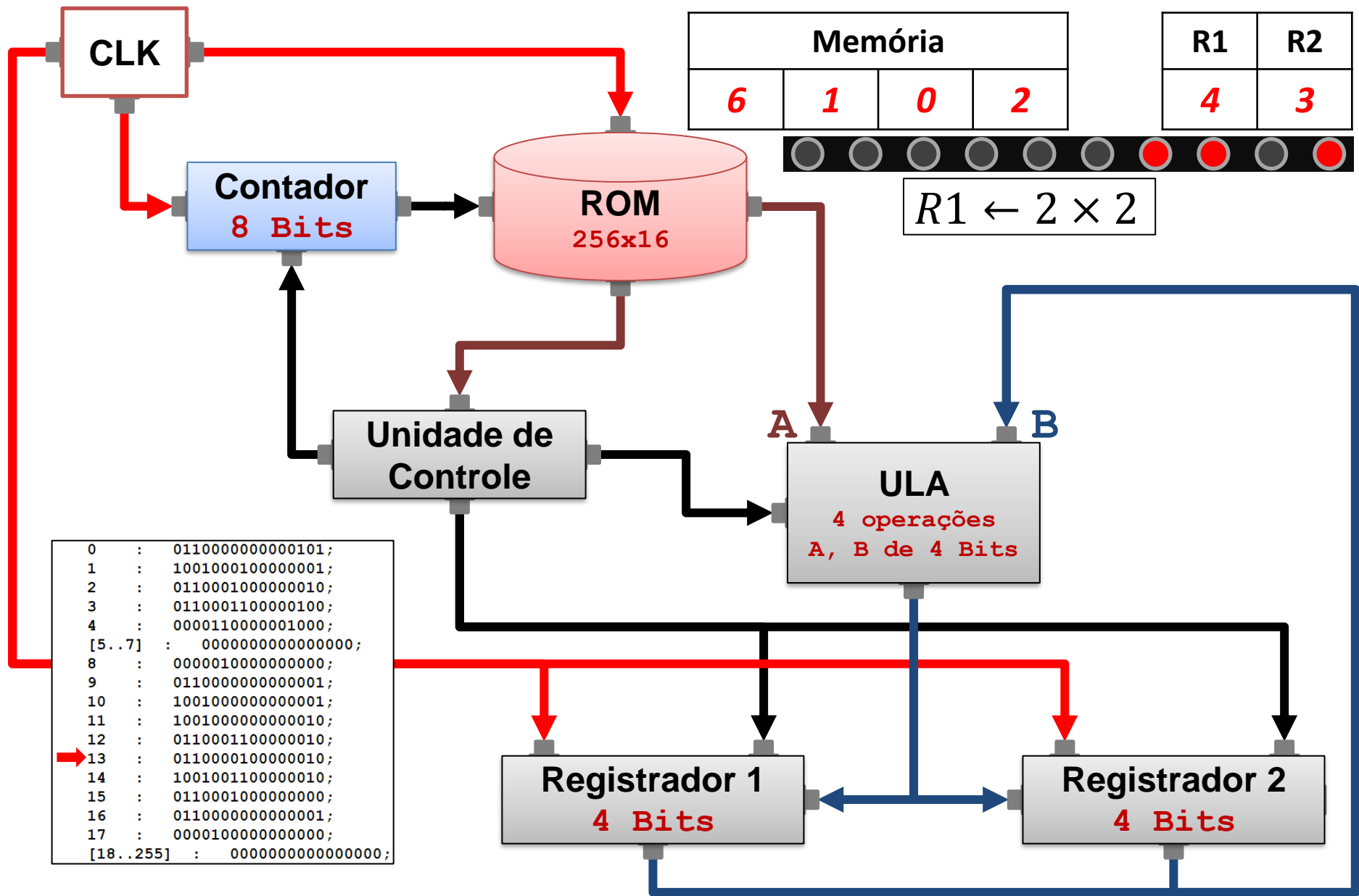


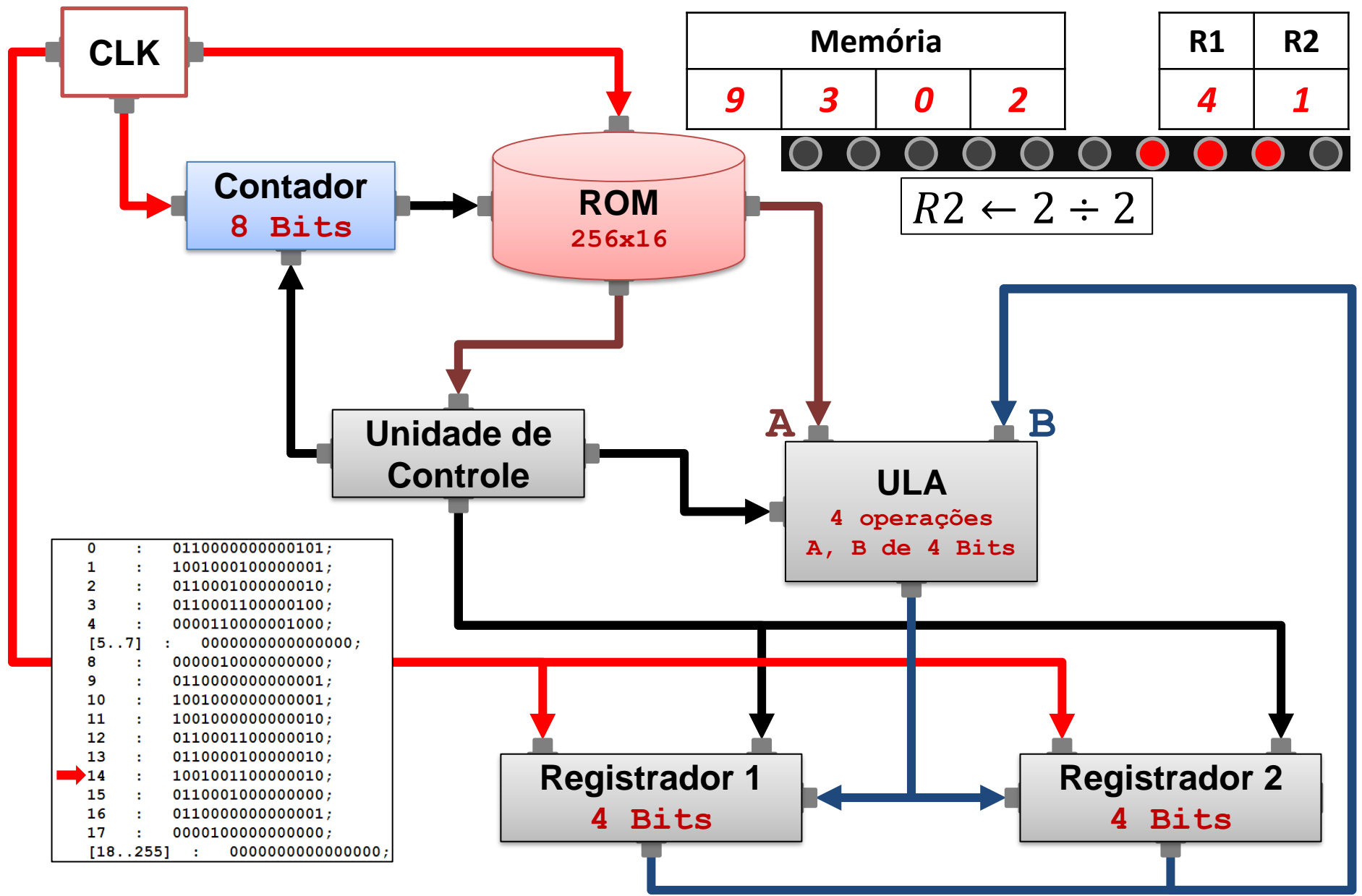


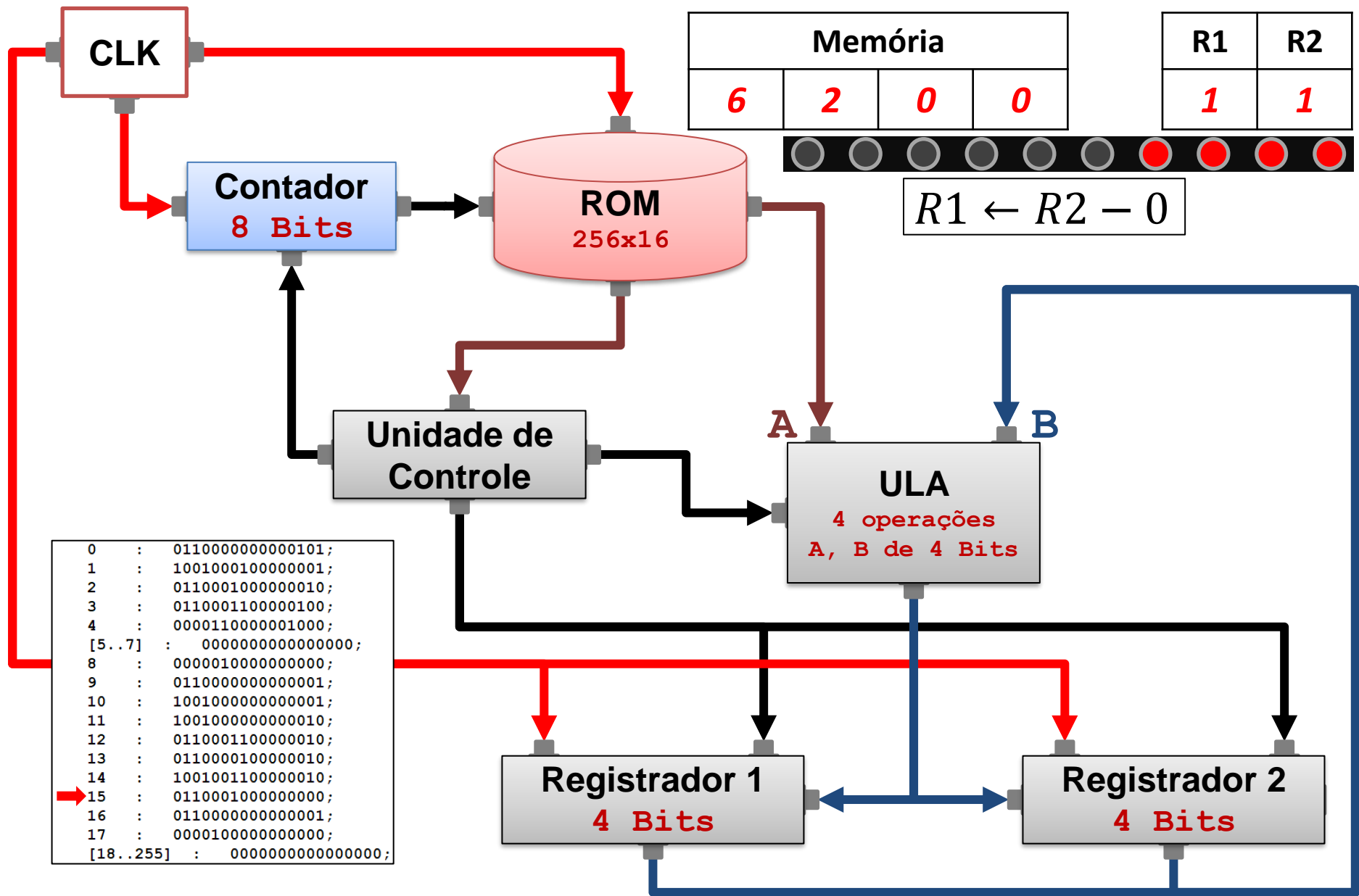


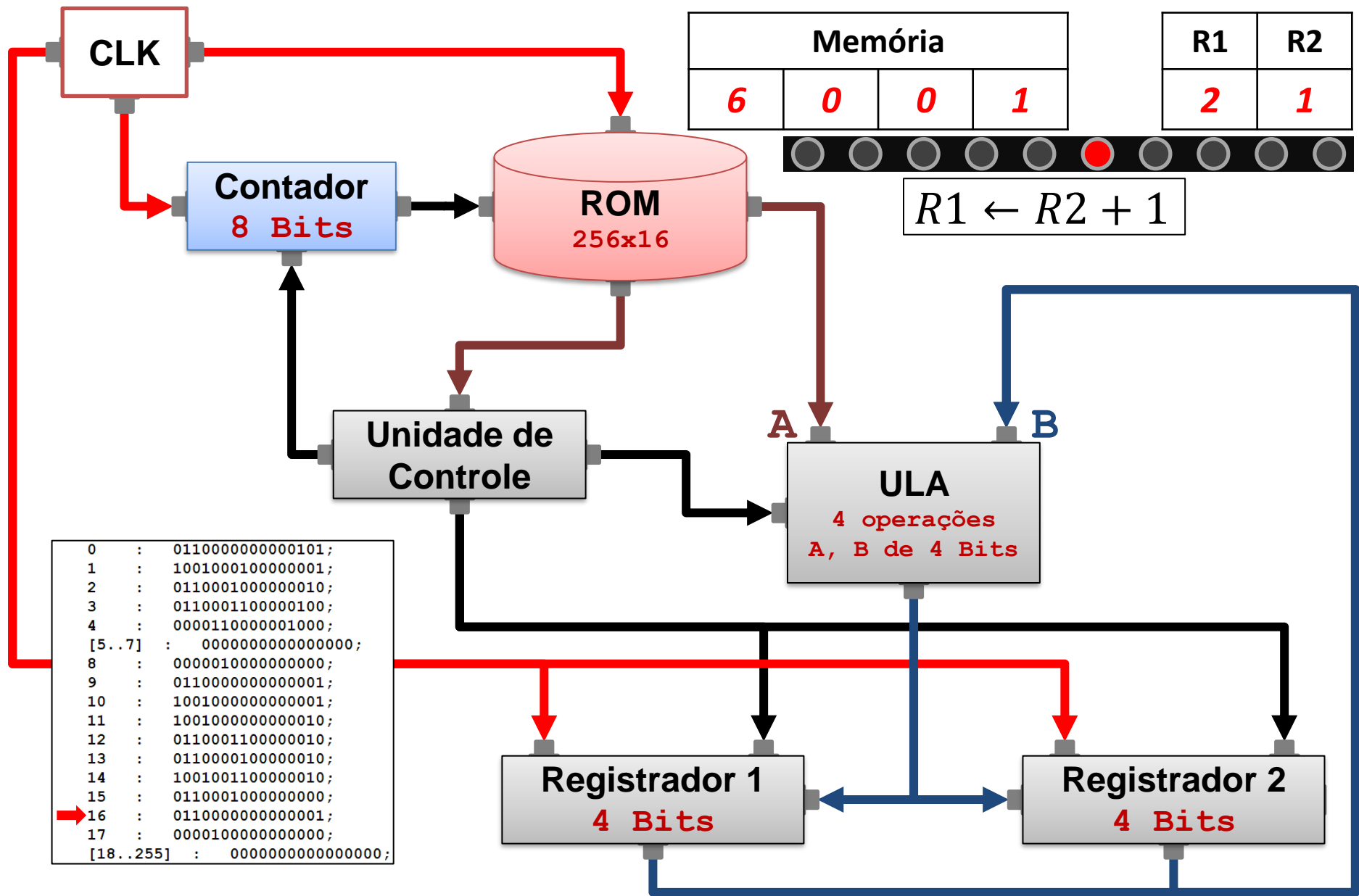


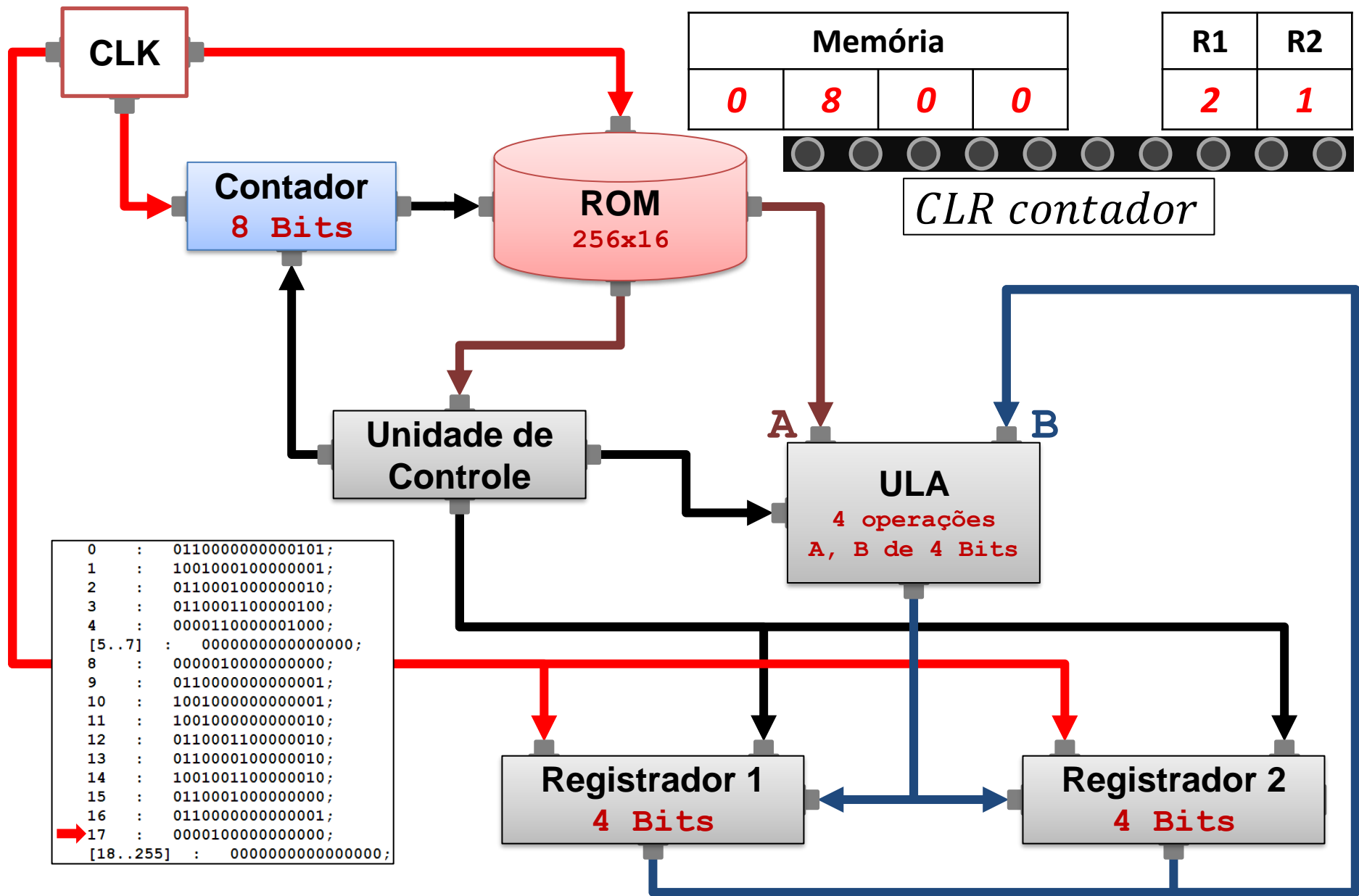


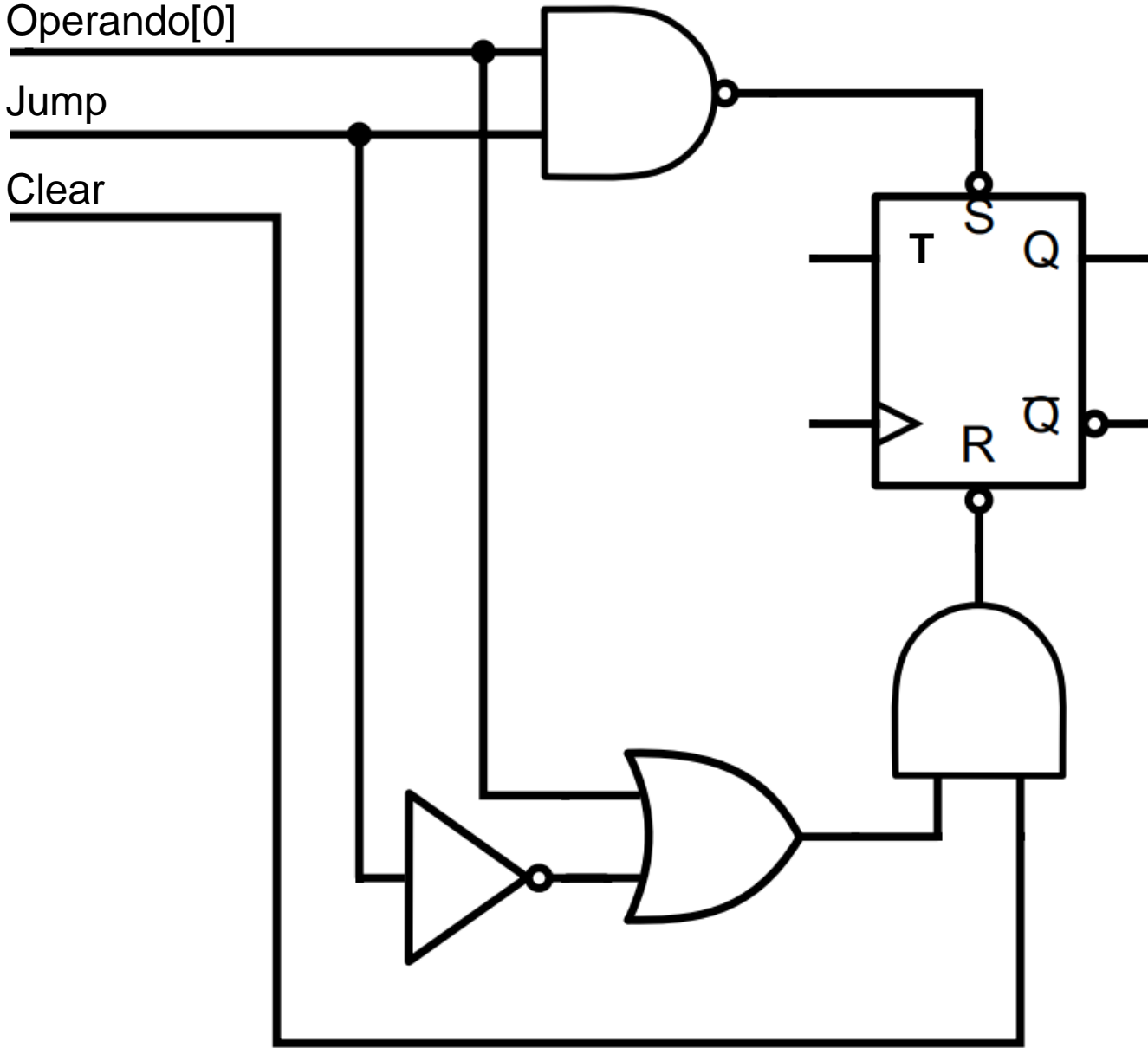








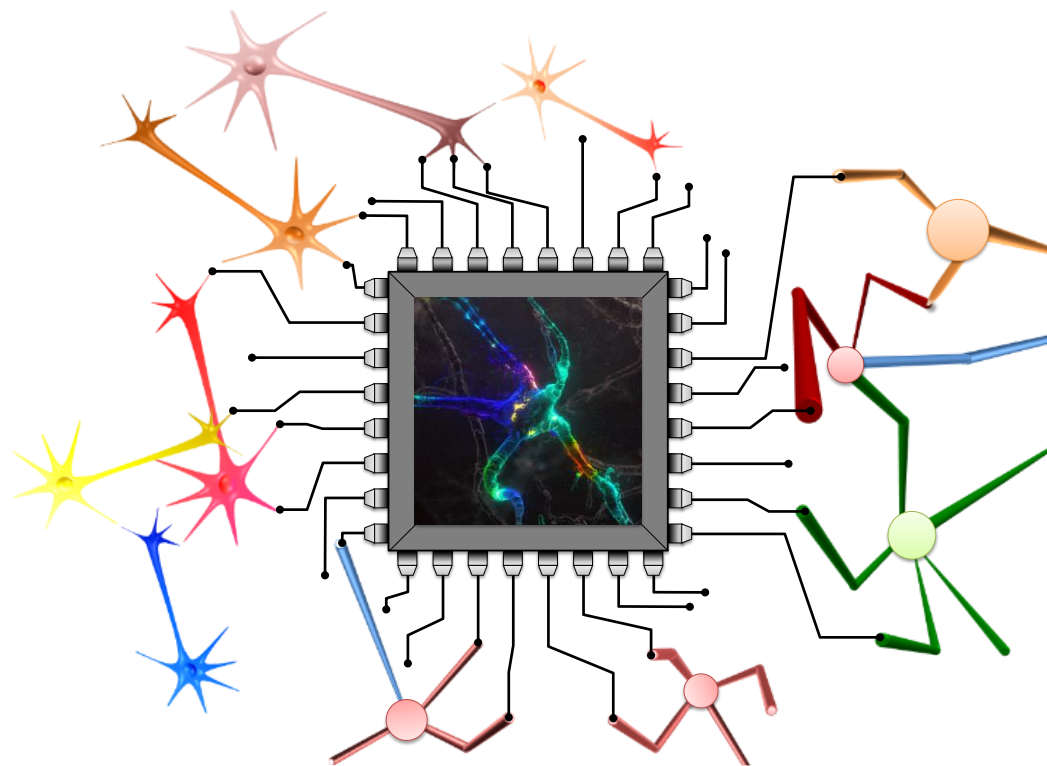




- Memória: Displays 7 seg **mais** significativos.
- Registradores: Displays 7 seg **menos** significativos.
- O **CLK** do circuito deverá ser de agora por diante associado à chave **mais significativa** disponível (SW9 por exemplo).
- O Master **CLR** do circuito deverá ser associado de agora por diante à chave **menos significativa** (SW0).

- Realizar as adaptações necessárias ao circuito para poder utilizar o JUMP do contador.
- Atentar para o fato de que o operando do JUMP é de 4 bits.
- Realizar as adaptações necessárias ao circuito para poder integrar os registradores na ULA.
- A lógica da Unidade de Controle deve ser **CENTRALIZADA**.

spatti@icmc.usp.br



GE4Bio – Grupo de Estudos em Sinais Biológicos