



Hoe werkt een rekenmachine?

- Uit welke hardware-componenten bestaat een rekenmachine?
- Welke instructies kan de machine uitvoeren?
- Practicum met de rekenmachine I
- Constante getallen
- Instructies van het type immediate
- Practicum met de rekenmachine II



Hoe werkt een rekenmachine?

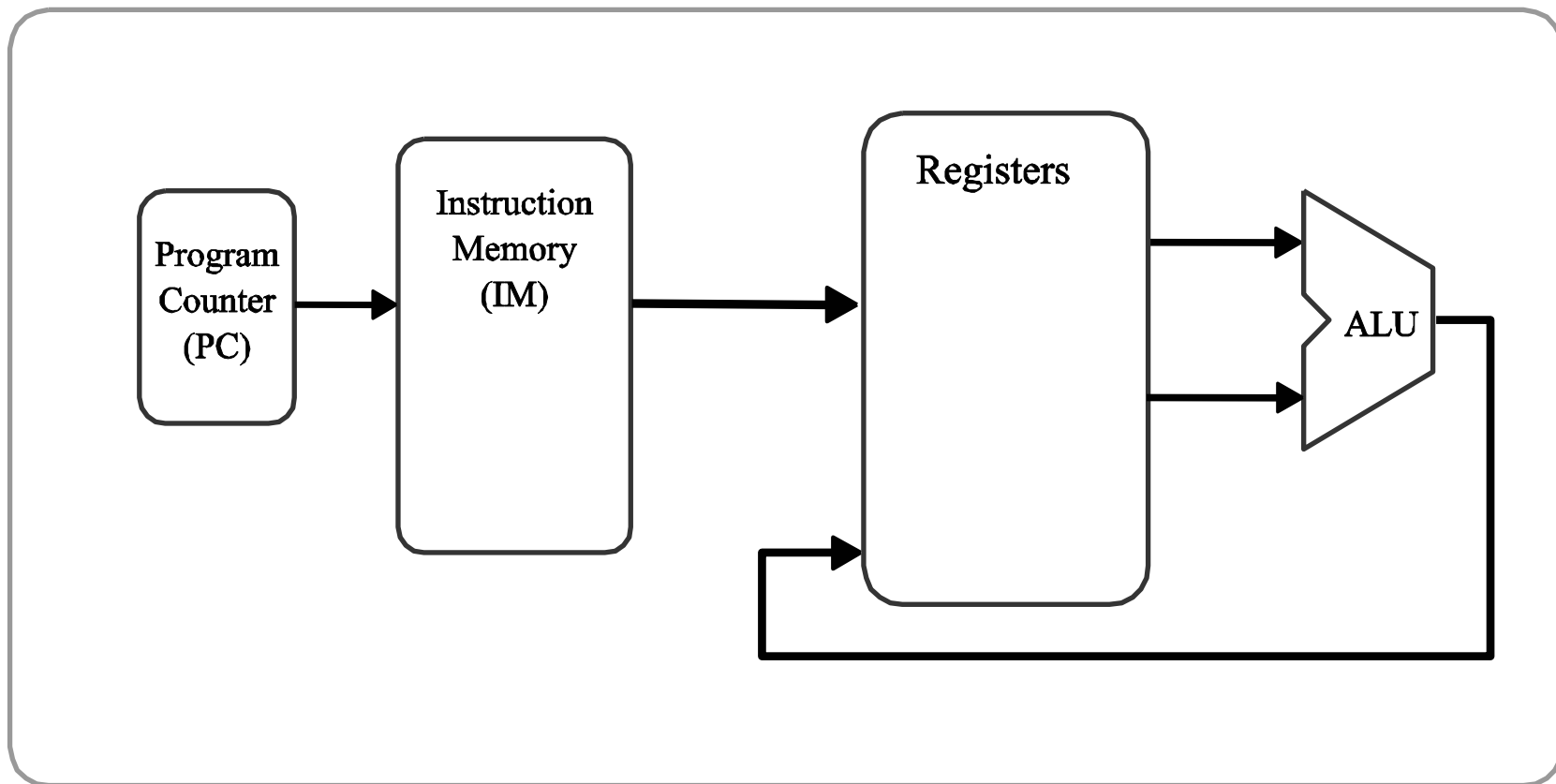
- Uit welke hardware-componenten bestaat een rekenmachine?
- Welke instructies kan de machine uitvoeren?
- Practicum met de rekenmachine I
- Constante getallen
- Instructies van het type immediate
- Practicum met de rekenmachine II



Rekenmachine: vier hoofdcomponenten:

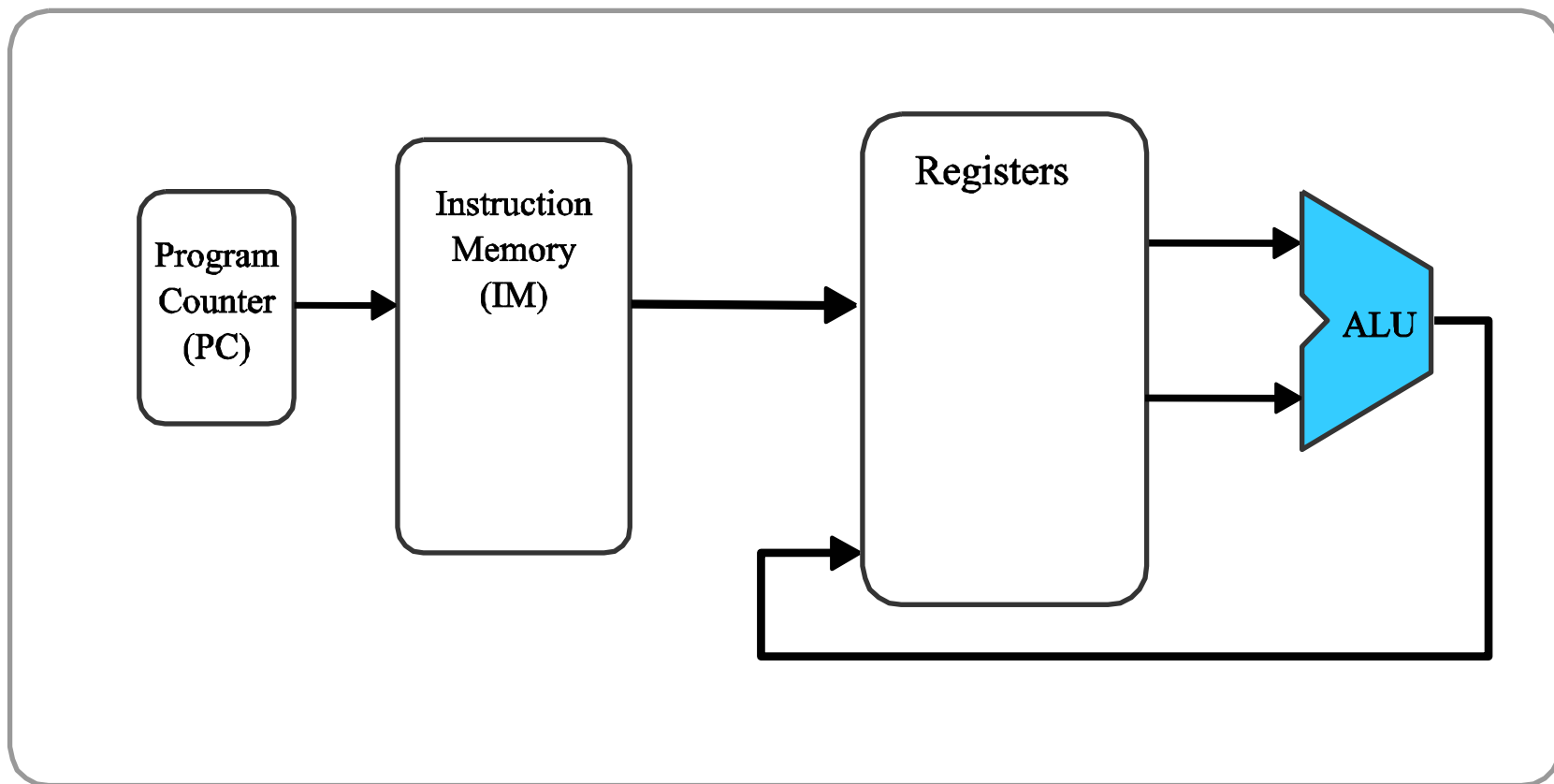
- Program Counter
- Instruction Memory
- Registers
- Arithmetic Logic Unit (ALU)

Samenhang tussen deze 4 hoofdcomponenten:



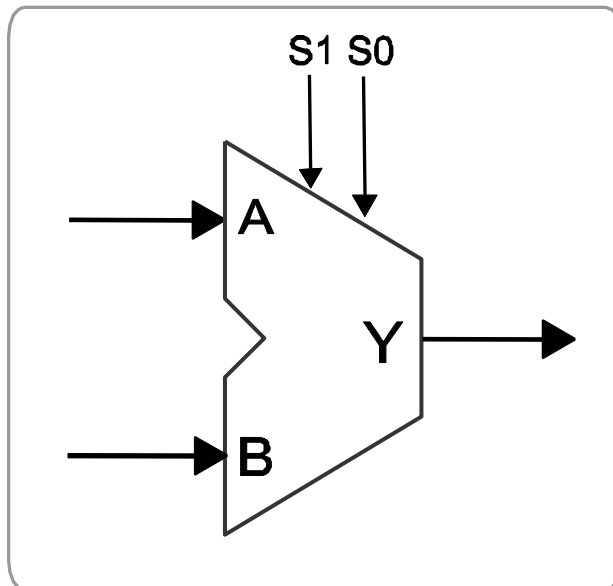
Architectuur van de rekenmachine

4 hoofdcomponenten:



Arithmetic Logic Unit: het rekenorgaan

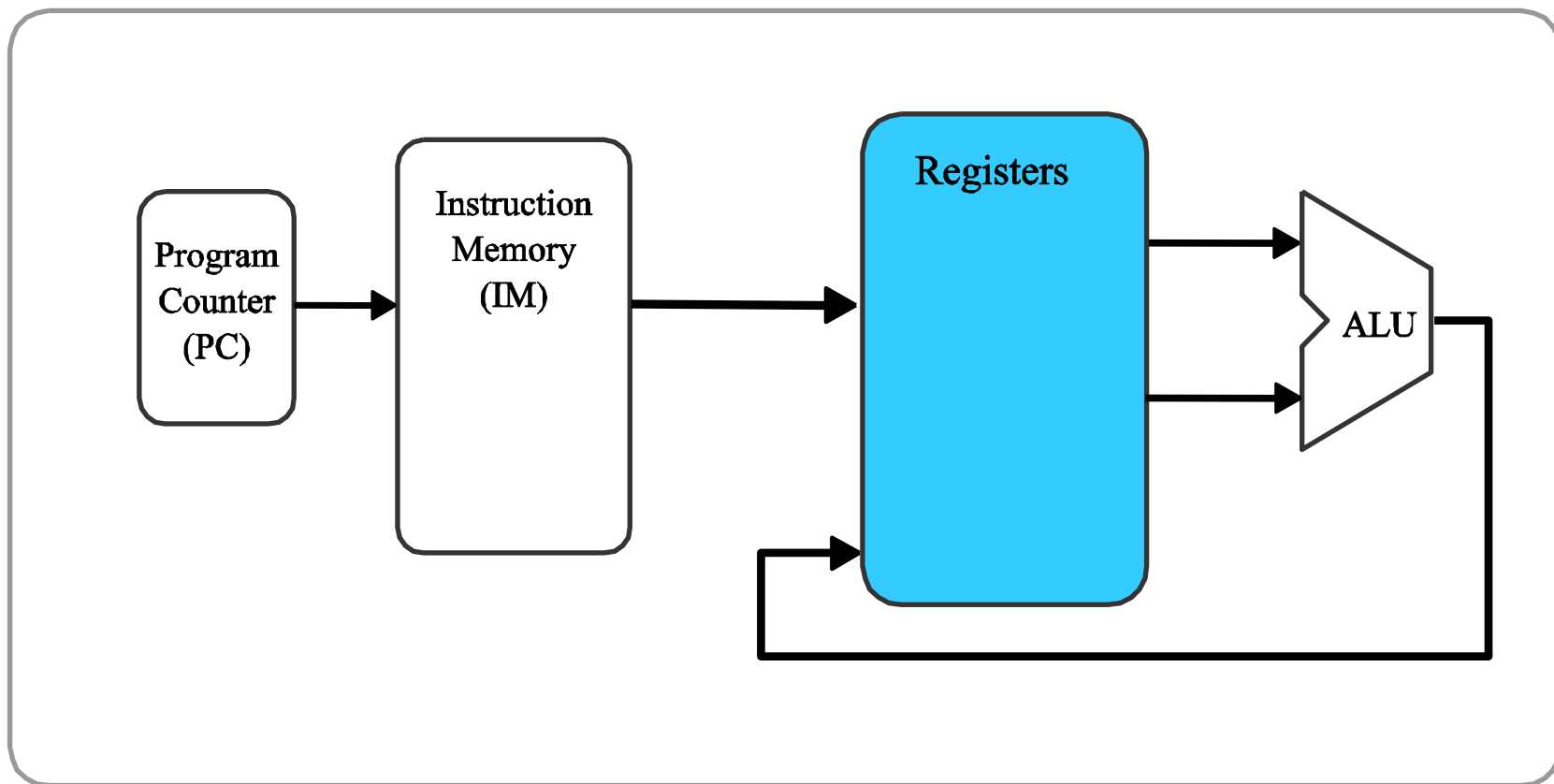
ALU



S1	S0	Operator	Functie
0	0	+	$Y = A + B$
0	1	-	$Y = A - B$
1	0	& (bitwise AND)	$Y = A \& B$
1	1	B wordt doorgelaten	$Y = B$

Architectuur van de rekenmachine

4 hoofdcomponenten:

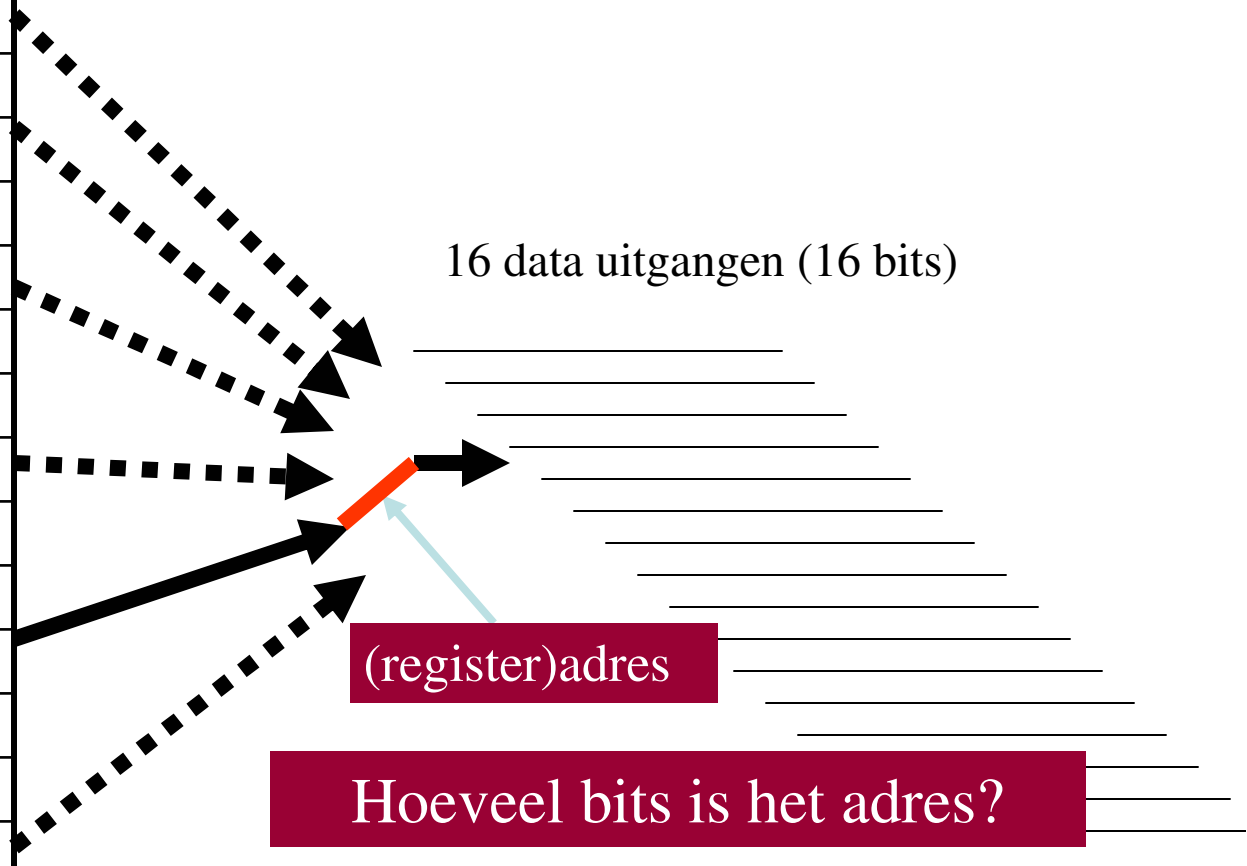


Opslag van (tussen)resultaten



adres		inhoud
0	0000	
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	1001000111000001
10	1010	
11	1011	
12	1100	0100100101100011
13	1101	
14	1110	
15	1111	

Registers



(register)adres

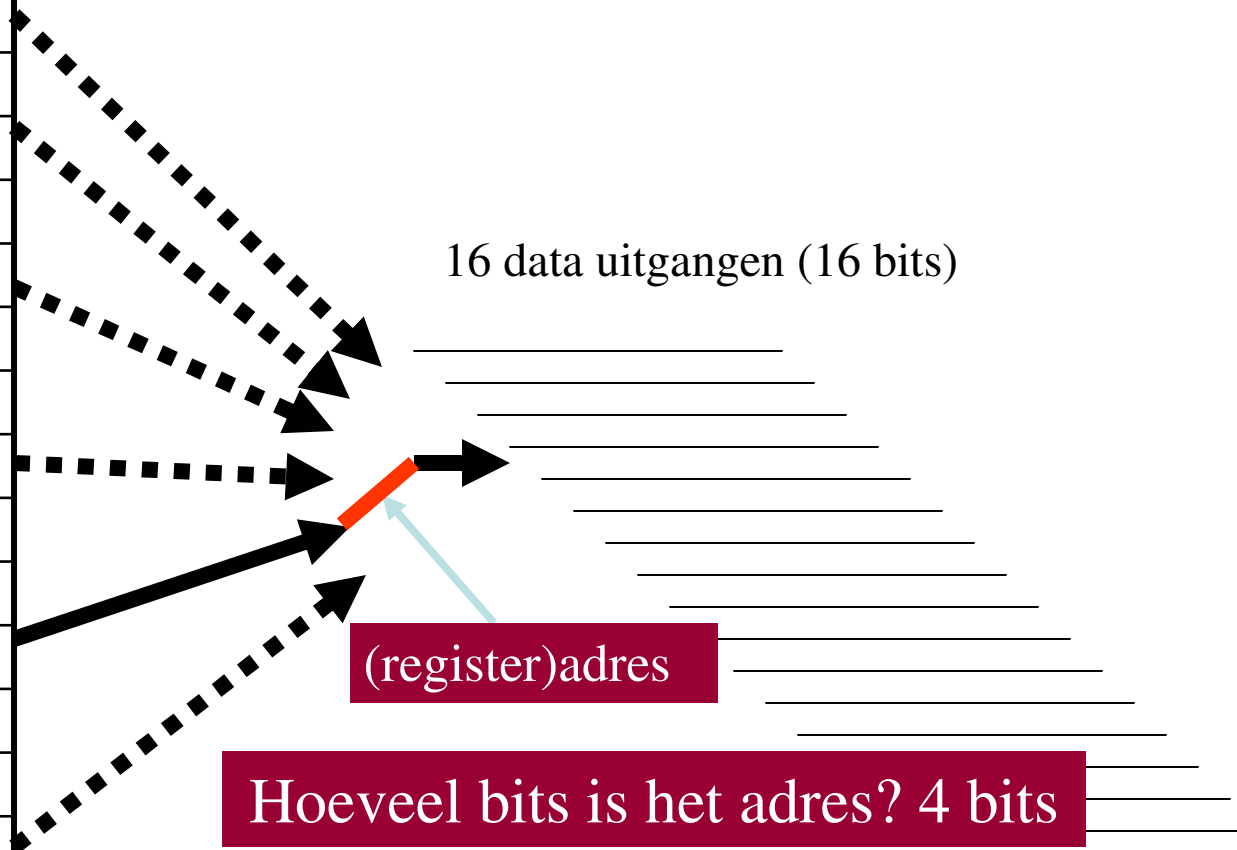
Hoeveel bits is het adres?

Hoeveel bits is dit geheugen?



adres	Inhoud (16 bits)	
0	0000	
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	1001000111000001
10	1010	
11	1011	
12	1100	0100100101100011
13	1101	
14	1110	
15	1111	

Registers



16 data uitgangen (16 bits)

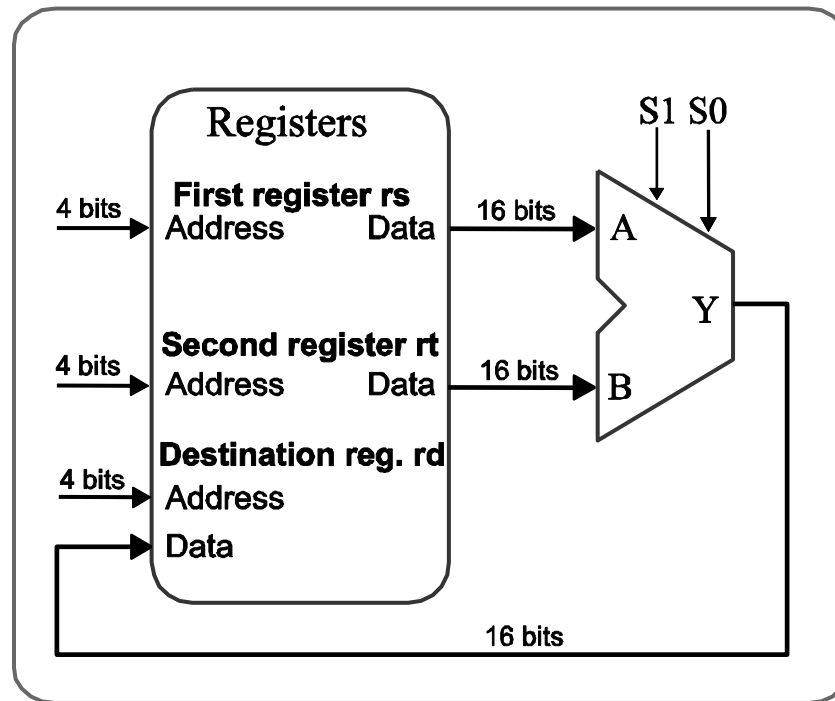
(register)adres

Hoeveel bits is het adres? 4 bits

Hoeveel bits is dit geheugen? $2^4 * 16 = 256$ bits

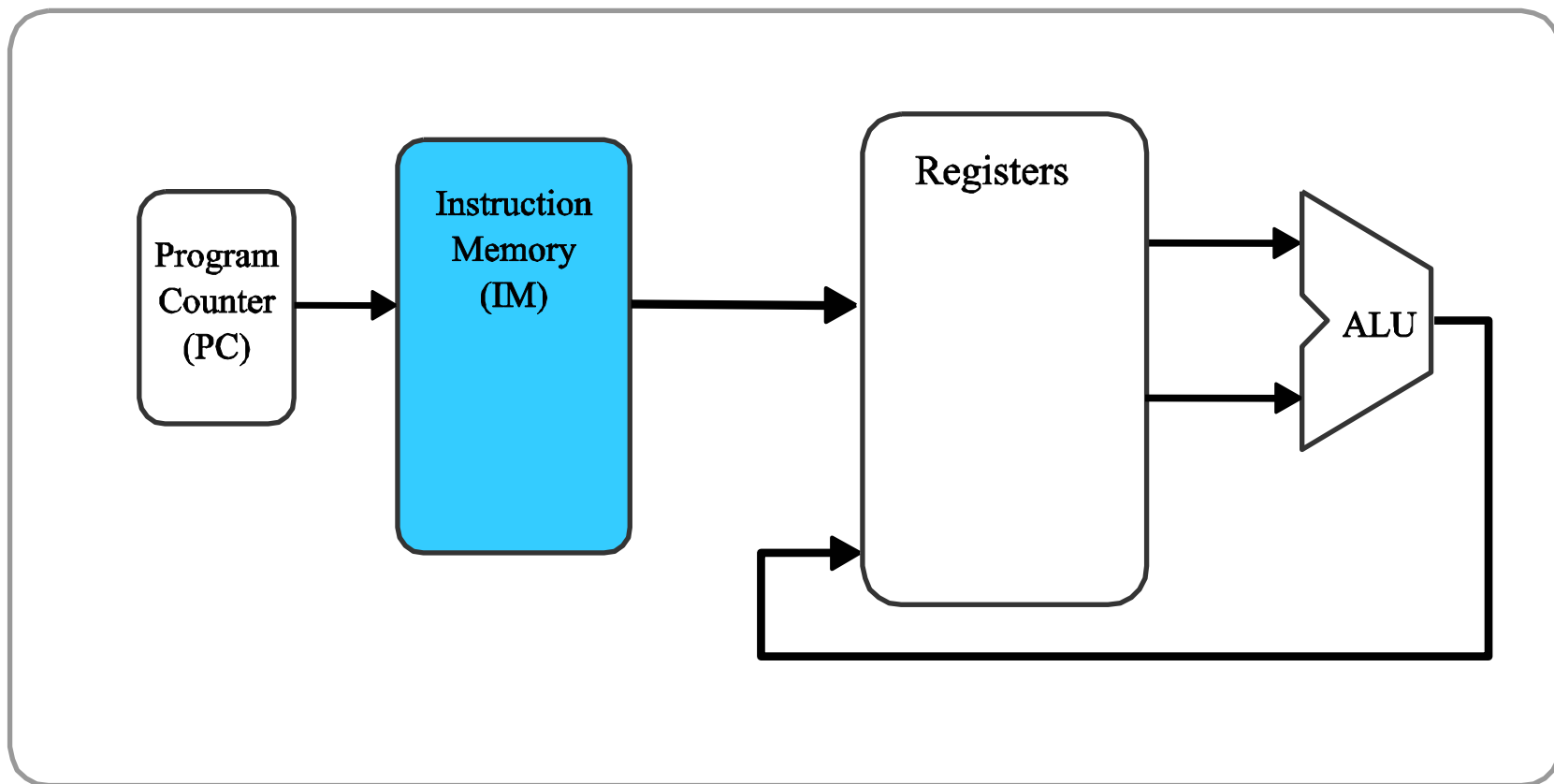


Datapad: Registers + ALU



Architectuur van de rekenmachine

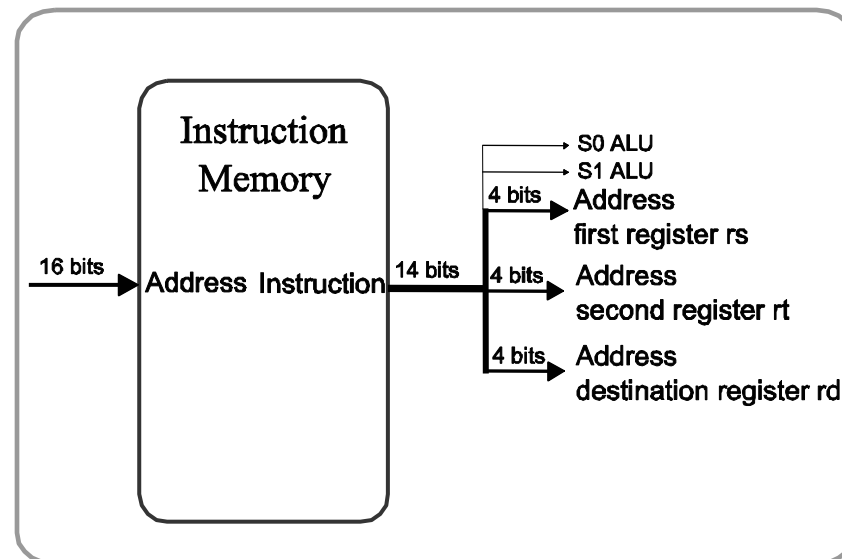
4 hoofdcomponenten:



Machinetaalprogramma; één geheugenplaats per instructie

Instruction Memory

Adres (16 bits)		Inhoud (14 bits) Machinecode
0	0000000000000000	10010001110010
1	0000000000000001	10010011110011
2	0000000000000010	10011001110010
3	0000000000000011	00010001110011
4	0000000000000100	
5	0000000000000101	
6	0000000000000110	
7	0000000000000111	
8	0000000000001000	
9	0000000000001001	
10	0000000000001010	

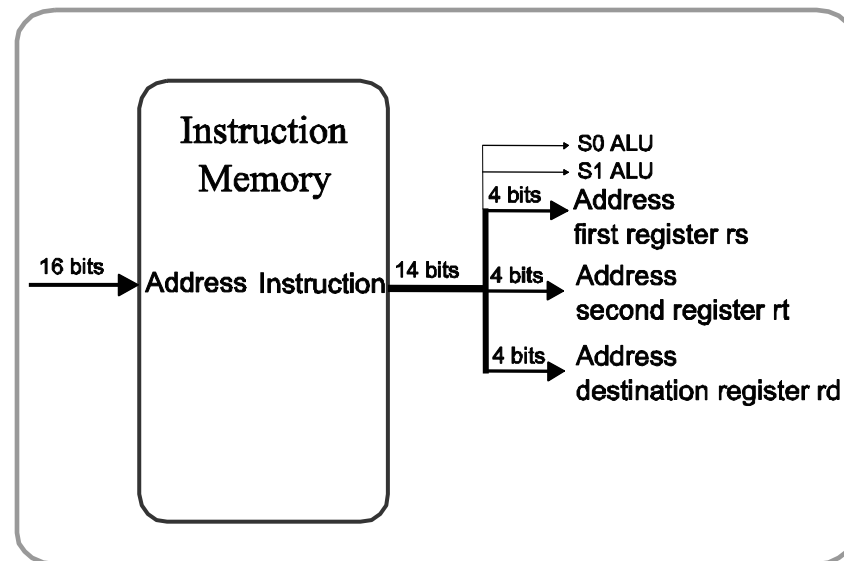


Hoeveel instructies kunnen we in dit geheugen opslaan?

Hoeveel bits is dit geheugen?

Instruction Memory

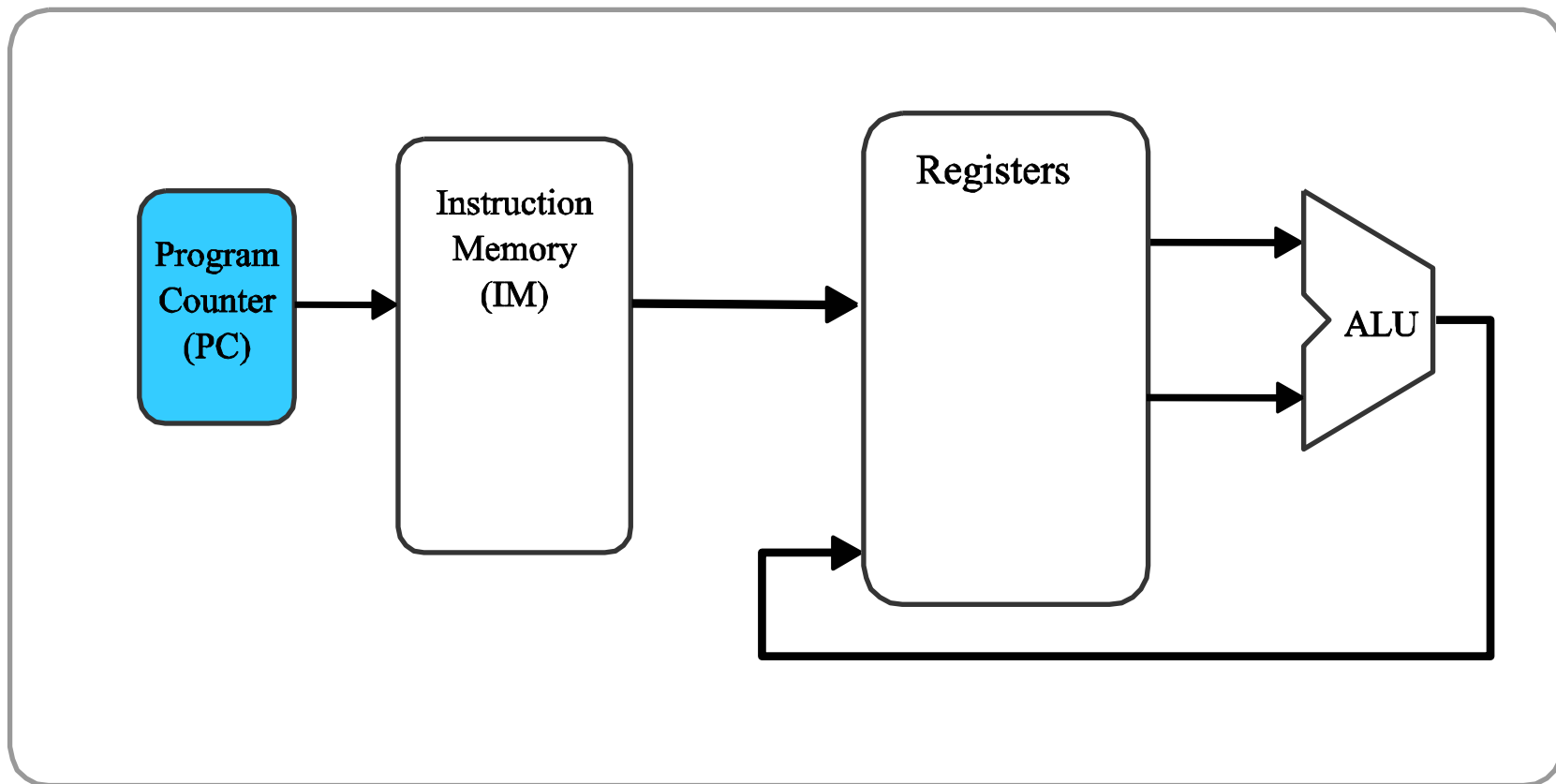
Adres (16 bits)	Inhoud (14 bits) Machinecode
0	0000000000000000 10010001110010
1	0000000000000001 10010011110011
2	0000000000000010 10011001110010
3	0000000000000011 00010001110011
4	0000000000000100
5	0000000000000101
6	0000000000000110
7	0000000000000111
8	0000000000001000
9	0000000000001001
10	0000000000001010



$2^{16} = 65.536$ instructies
 $2^{16} \times 14 = 65.536 \times 14 = 917.504$ bits

Architectuur van de rekenmachine

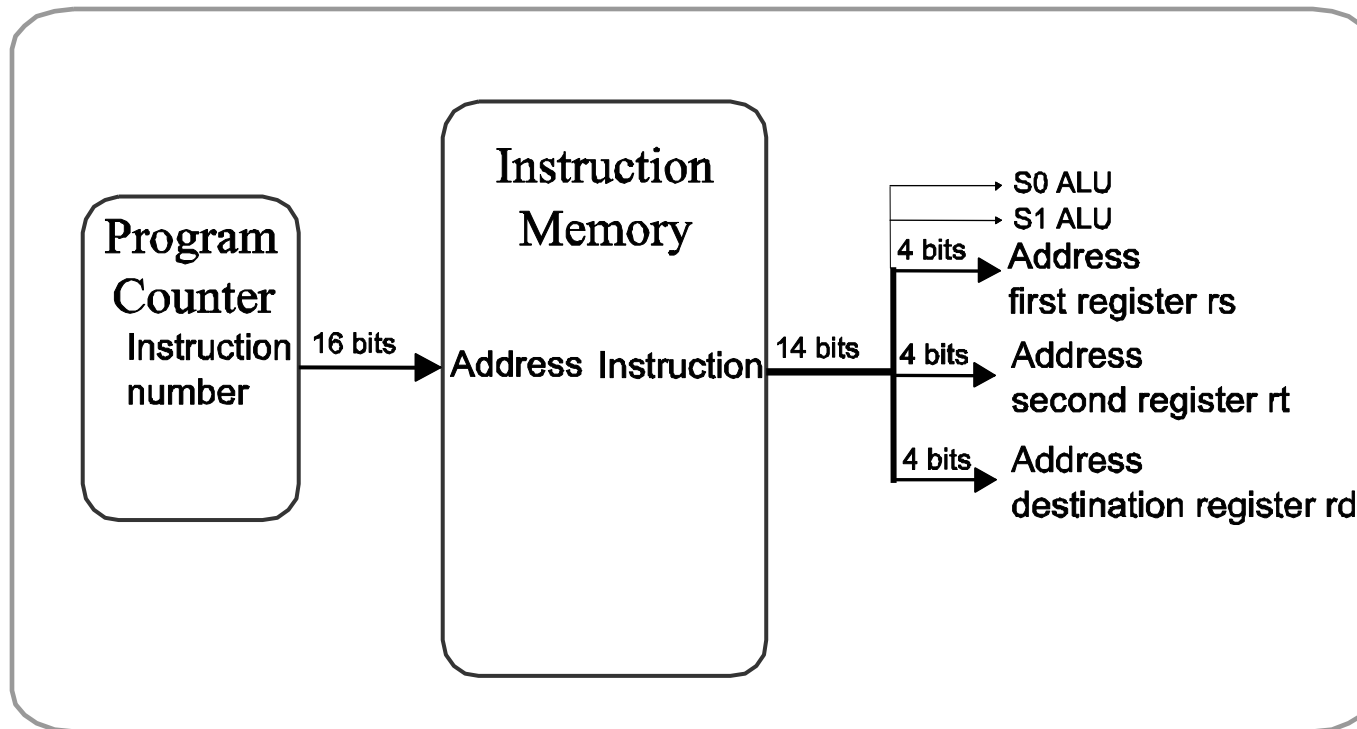
4 hoofdcomponenten:



Houdt bij welke instructie wordt uitgevoerd



Instructiepad: PC + Instruction Memory

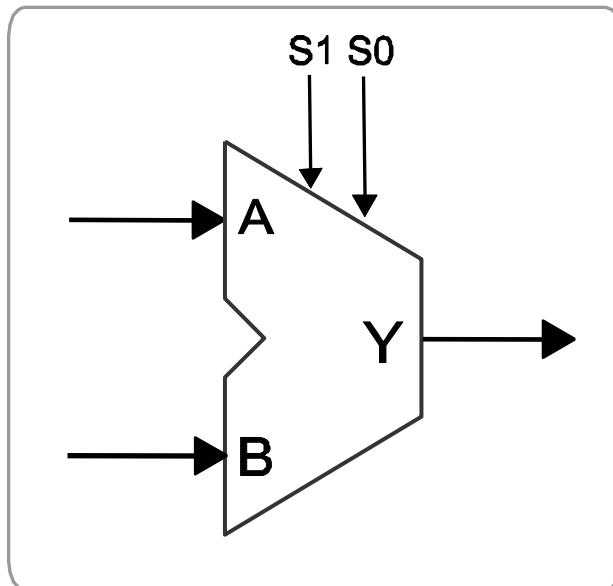




Hoe werkt een rekenmachine?

- Uit welke hardware-componenten bestaat een rekenmachine?
- **Welke instructies kan de machine uitvoeren?**
- Practicum met de rekenmachine I
- Constante getallen
- Instructies van het type immediate
- Practicum met de rekenmachine II

Terug naar de ALU



S1	S0	Operator	Functie
0	0	+	$Y = A + B$
0	1	-	$Y = A - B$
1	0	& (bitwise AND)	$Y = A \& B$
1	1	B wordt doorgelaten	$Y = B$



Instructies

S1	S0	Operator	Instructie	Functie
0	0	+	ADD	$Y = A + B$
0	1	-	SUB	$Y = A - B$
1	0	& (bitwise AND)	AND	$Y = A \& B$
1	1	B wordt doorgelaten	COPY	$Y = B$



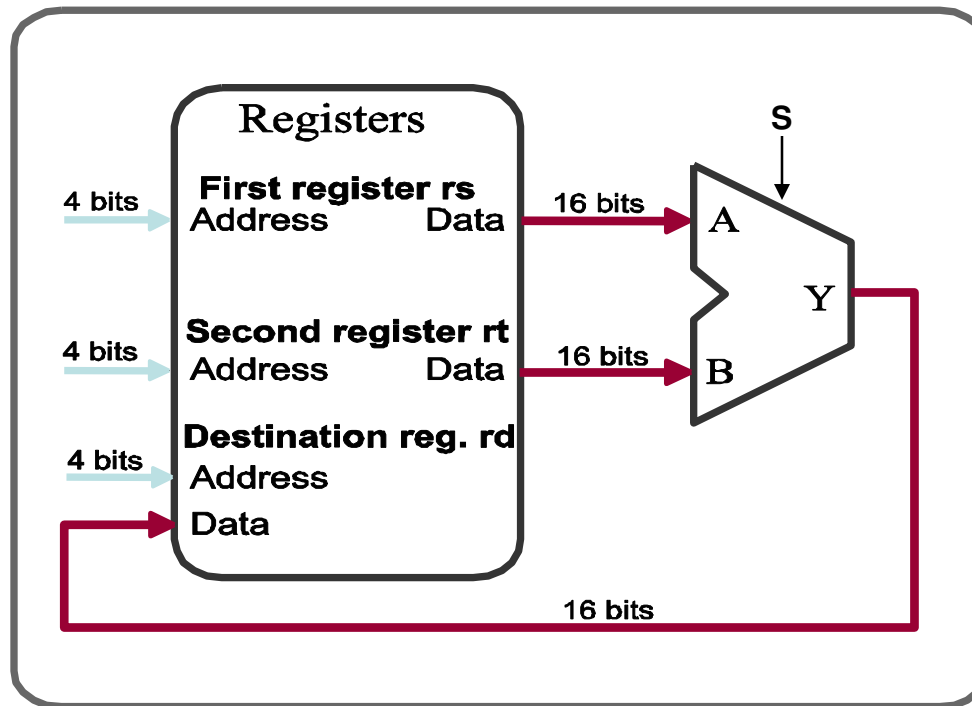
Instructies

S1	S0	Operator	Instructie	Functie
0	0	+	ADD	$Y = A + B$
0	1	-	SUB	$Y = A - B$
1	0	& (bitwise AND)	AND	$Y = A \& B$
1	1	B wordt doorgelaten	COPY	$Y = B$

Instructie	Betekenis	Voorbeeld	Betekenis
ADD rd, rs, rt	Optellen registers	ADD \$7, \$3, \$4	$r7 \leftarrow r3 + r4$
SUB rd, rs, rt	Aftrekken registers	SUB \$7, \$3, \$4	$r7 \leftarrow r3 - r4$
AND rd, rs, rt	Bitwise-AND registers	AND \$7, \$3, \$4	$r7 \leftarrow r3 \& r4$
COPY rd, rt	Kopieer register	COPY \$7, \$4	$r7 \leftarrow r4$

Instructieset rekenmachine

Assembly Language



Syntax: **ADD** rd, rs, rt

Voorbeeld: **ADD** \$7, \$5, \$6

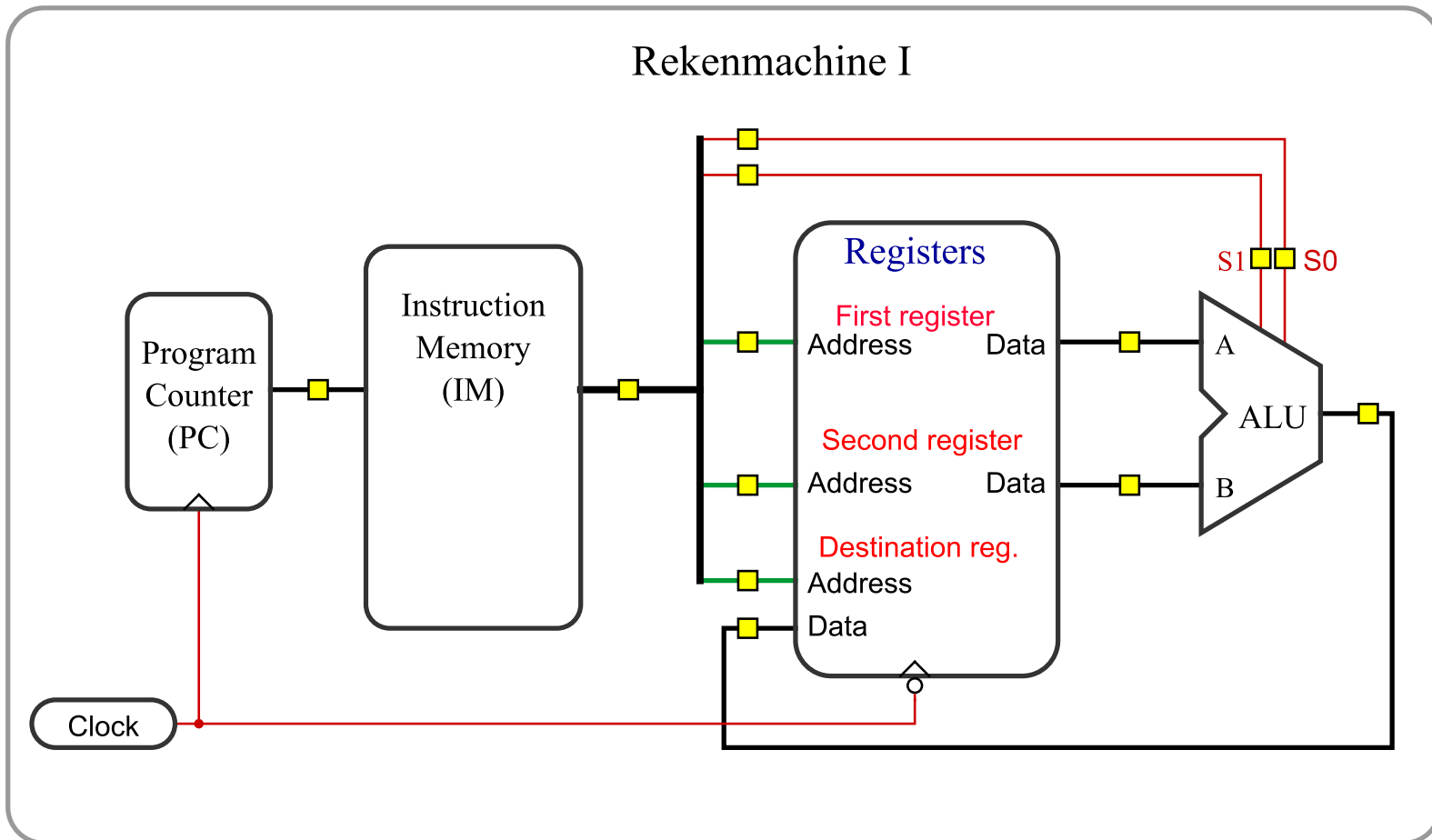
Betekenis: het getal in register 7 \leftarrow het getal in register 5 + het getal in register 6



Belangrijke abstractie: variabelen zijn registeradressen!

- $c \leftarrow a + b$ // c wordt a + b
- $rd \leftarrow rs + rt$
- ADD \$7, \$5, \$6
- Het getal in register 7 wordt het getal in register 5 + het getal in register 6

Rekenmachine I



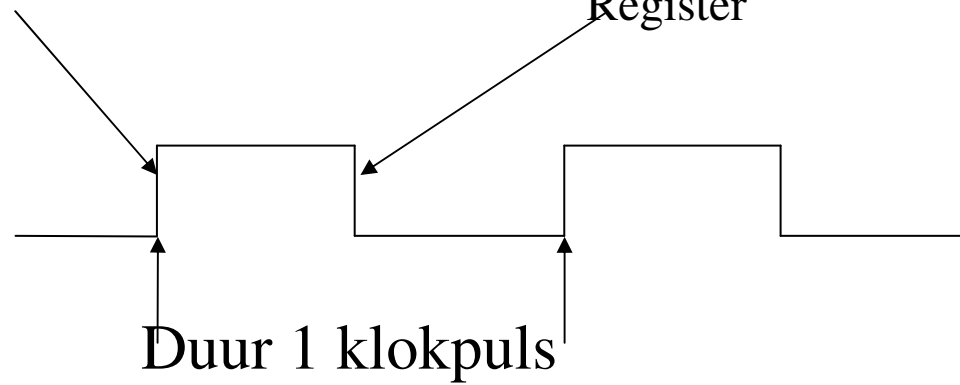


Timing

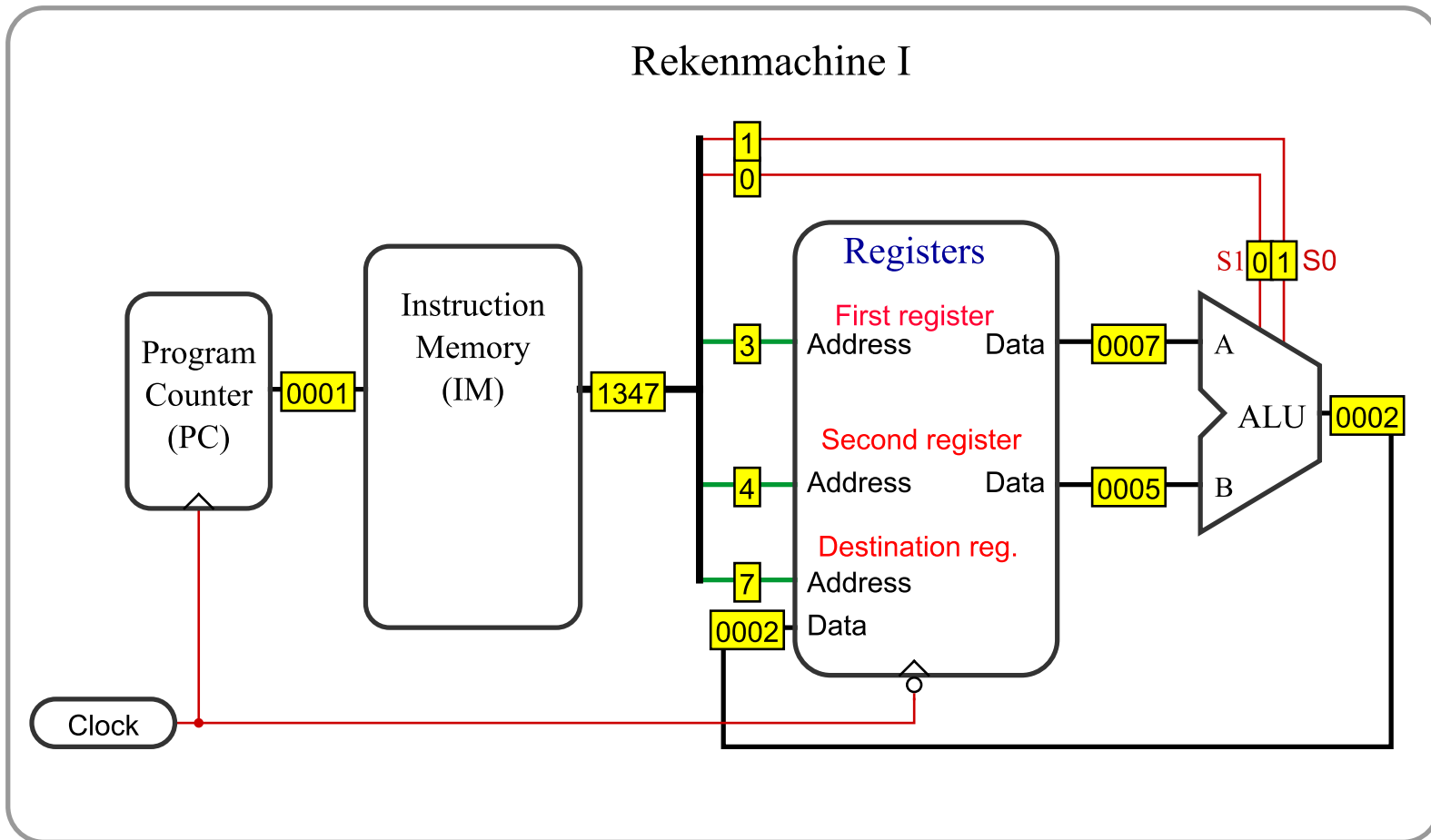
Positieve klokflank
 $PC = PC + 1$

Negatieve klokflank
Data in Destination
Register

kloksignaal



Rekenmachine I



SUB \$7, \$3, \$4

7 - 5 = 2



Hoe werkt een rekenmachine?

- Uit welke hardware-componenten bestaat een rekenmachine?
- Welke instructies kan de machine uitvoeren?
- **Practicum met de rekenmachine I**
- Constante getallen
- Instructies van het type immediate
- **Practicum met de rekenmachine II**



Practicum met de Rekenmachine I

Maak de opdrachten van paragraaf 3.4.



Hoe werkt een rekenmachine?

- Uit welke hardware-componenten bestaat een rekenmachine?
- Welke instructies kan de machine uitvoeren?
- Practicum met de rekenmachine I
- **Constante getallen**
- Instructies van het type immediate
- Practicum met de rekenmachine II



Constance getallen

- Waarom zijn constante getallen zo belangrijk?
 - Heel veel instructies bevatten een constant getal
- Waar worden constante getallen opgeslagen?
 - In het instructiegeheugen
- Welke type instructies werken met constante getallen?
 - Immediate instructies



Hoe werkt een rekenmachine?

- Uit welke hardware-componenten bestaat een rekenmachine?
- Welke instructies kan de machine uitvoeren?
- Practicum met de rekenmachine I
- Constante getallen
- **Instructies van het type immediate**
- Practicum met de rekenmachine II



Instructies van het type immediate

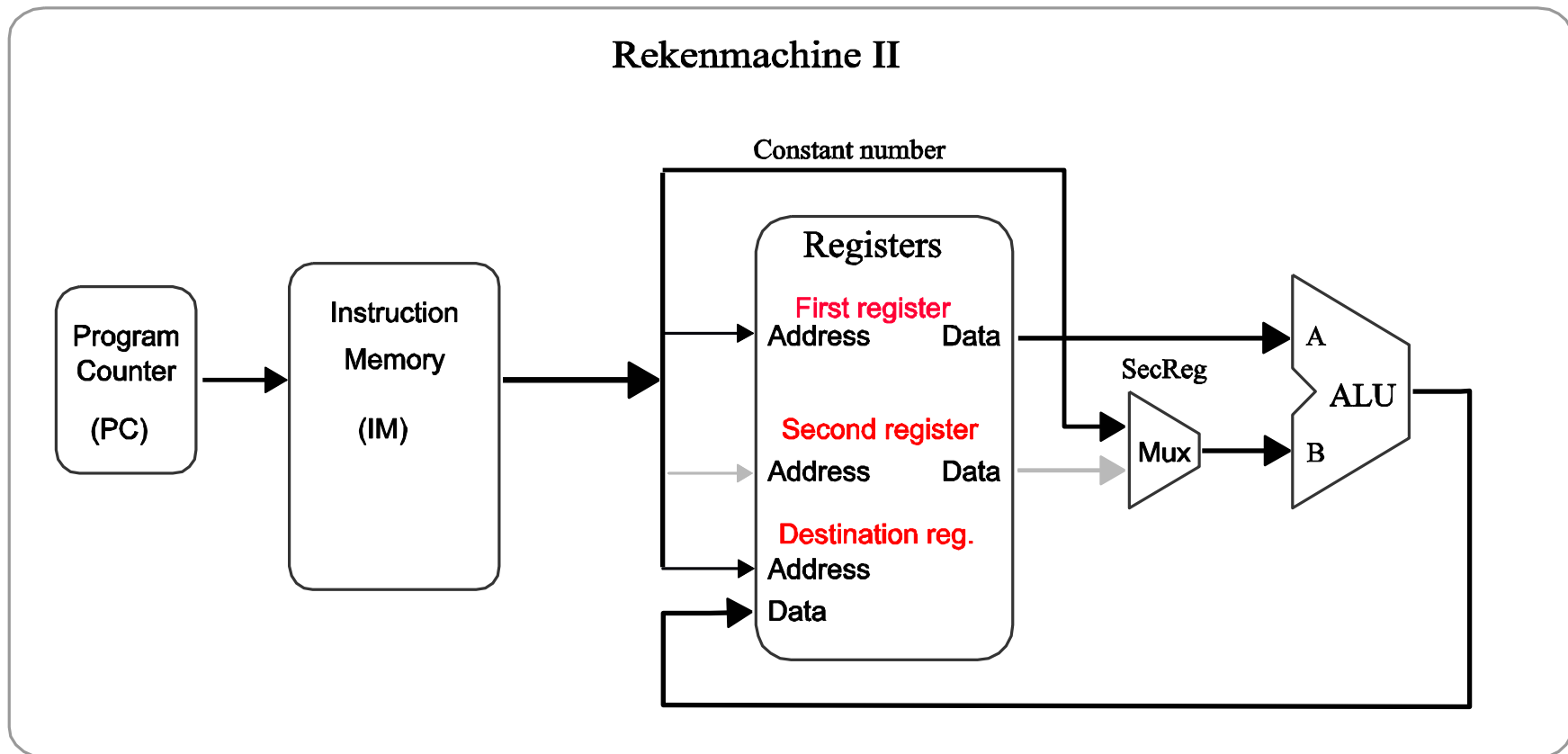
Rekenmachine II kent vier instructies van het type immediate:

1. ADDI (ADD Immediate)
2. SUBI
3. ANDI
4. LOADI

Bij de instructie ADDI wordt een constante opgeteld bij een variabele.

Bij de instructie LOADI wordt een constante opgehaald uit het instructiegeheugen.

Datapad van een immediate-instructie



Welke operatie voert de ALU uit bij een LOADI-instructie?



Instructietypen

- Rekenkundige en logische instructies
 - ADD
 - SUB
 - AND
- Immediate instructies
 - LOADI
 - ADDI
 - SUBI
 - ANDI
- Datatransfer
 - COPY

Instructieset van Rekenmachine II

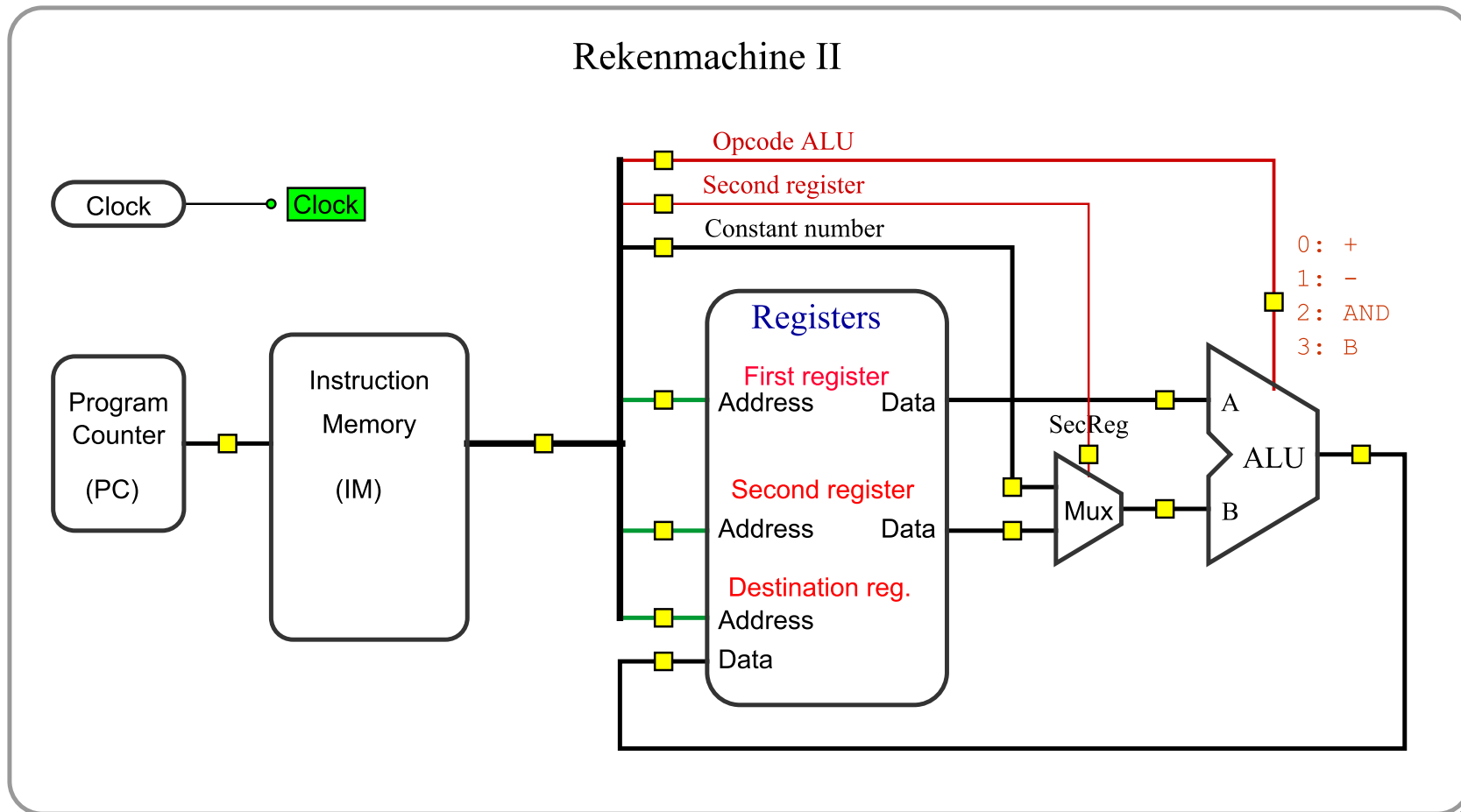
Instructie	Betekenis	Voorbeeld	Betekenis
ADD rd, rs, rt	Optellen registers	ADD \$5, \$6, \$7	$r5 \leftarrow r6 + r7$
SUB rd, rs, rt	Aftrekken registers	SUB \$5, \$6, \$7	$r5 \leftarrow r6 - r7$
AND rd, rs, rt	Bitwise AND registers	AND \$5, \$6, \$7	$r5 \leftarrow r6 \& r7$
COPY rd, rt	Copy register	COPY \$3, \$2	$r3 \leftarrow r2$
ADDI rd, rs, imm	Optellen register en constante	ADDI \$5, \$6, 0x1234	$r5 \leftarrow r6 + 0x1234$
SUBI rd, rs, imm	Aftrekken register en constante	SUBI \$7, \$6, 0x1234	$r7 \leftarrow r6 - 0x1234$
ANDI rd, rs, imm	Bitwise AND register en const.	ANDI \$5, \$6, 0d34	$r5 \leftarrow r6 \& 0d34$
LOADI rd, imm	Laad getal in register	LOAD \$1, 0x 0020	$r1 \leftarrow 0x0020$



Hoe werkt een rekenmachine?

- Uit welke hardware-componenten bestaat een rekenmachine?
- Welke instructies kan de machine uitvoeren?
- Practicum met de rekenmachine I
- Constante getallen
- Instructies van het type immediate
- **Practicum met de rekenmachine II**

Rekenmachine II





Practicum met de Rekenmachine II

Maak de opdrachten van paragraaf 3.6 en 3.7.