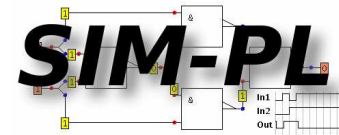


Hoe werkt een computer precies?

Met steun van stichting Edict

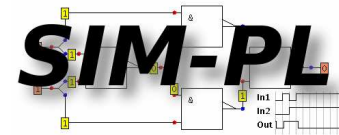


Ben Bruidegom & Wouter Koolen-Wijkstra
AMSTEL Instituut Universiteit van Amsterdam



Overview

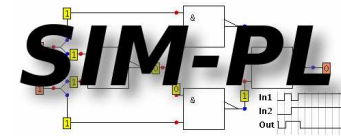
- Introductie SIM-PL
- Simulatietool voor werking computer
 - Rekenmachine
 - Uitbreiding: “loopjes”
 - Uitbreiding: datageheugen (Harvard)
- Ontwerptool voor schakelingen
 - Specificatie: logica
 - Implementatie: poorten
- SIM-PL in het VO



Introductie SIM-PL

- Software
 - Executer (Simulator)
 - Editor
- Bibliotheek modellen
 - Van poort tot processor
- Lesmateriaal
 - Presentaties
 - Practica

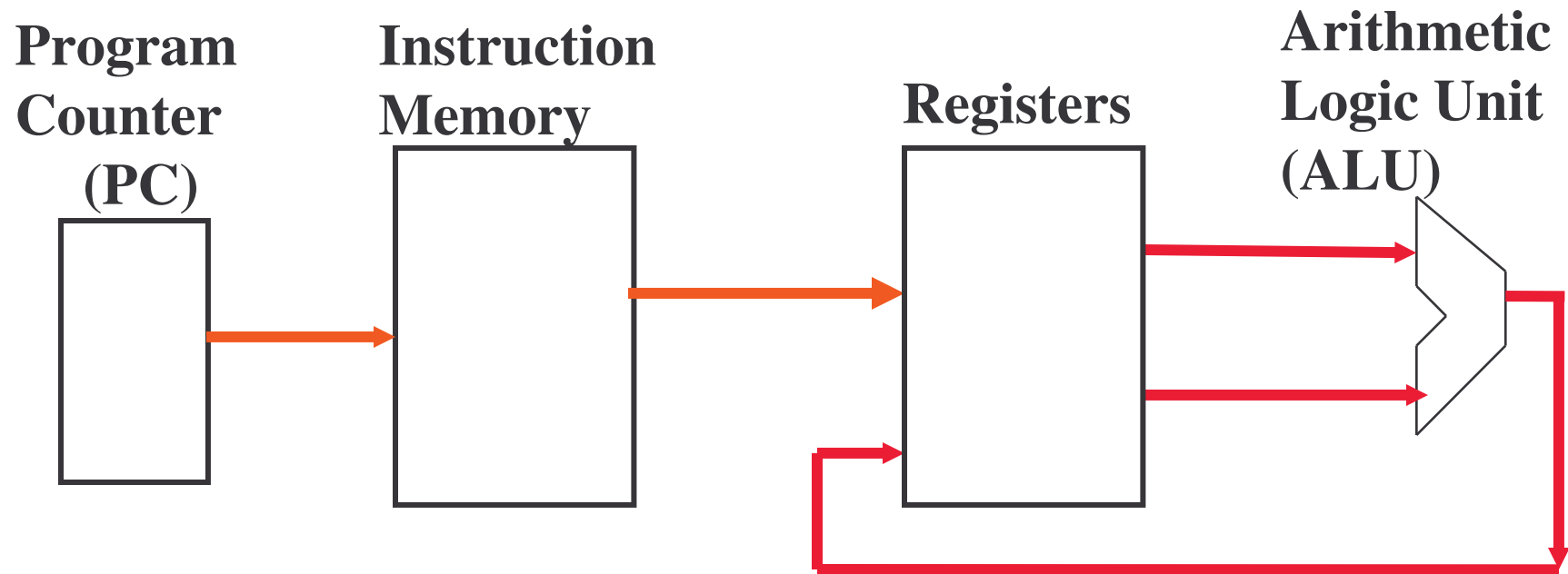
<http://www.science.uva.nl/amstel/SIM-PL>



Voorbeeld lesmateriaal

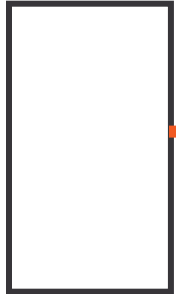
Hoe werkt een rekenmachine?

Rekenmachine bestaat uit 4 hoofdcomponenten:



ALU

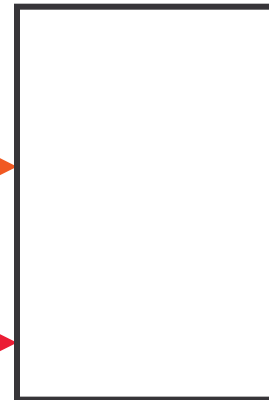
Program Counter (PC)



Instruction Memory



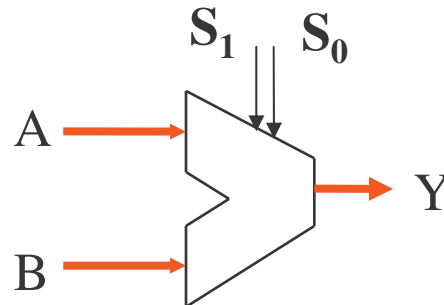
Registers



Arithmetic Logic Unit (ALU)

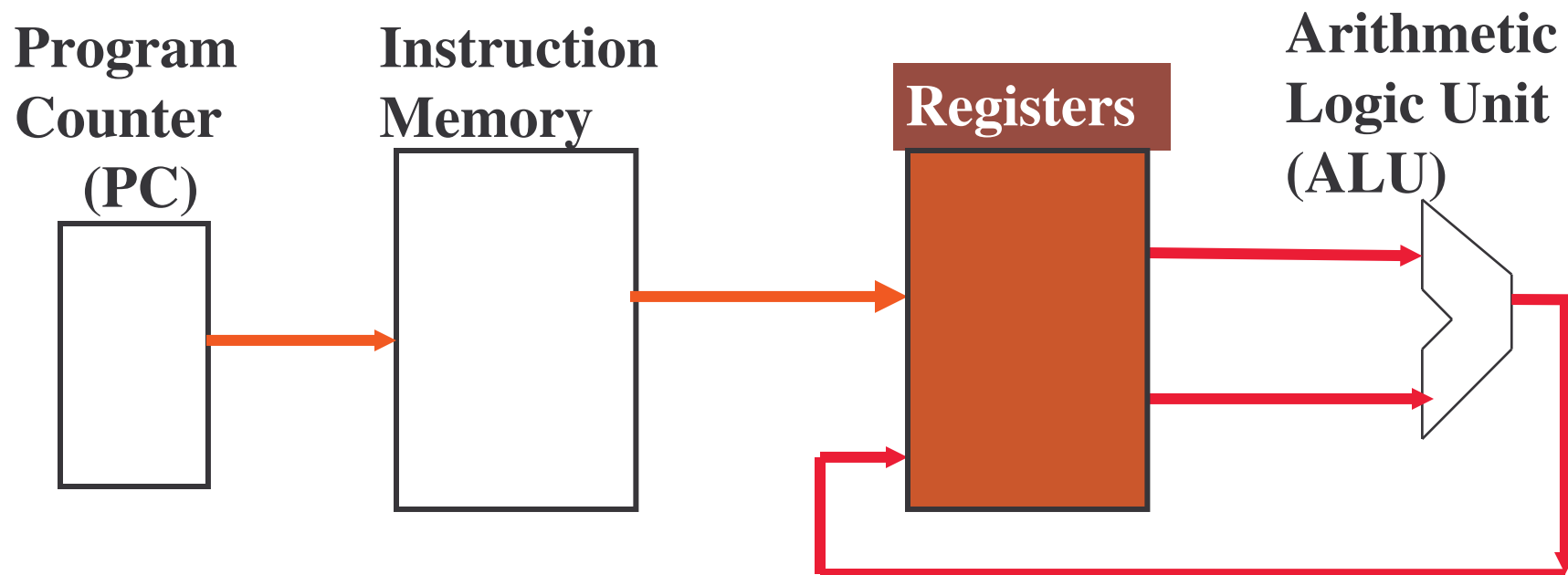


ALU (16 bit)

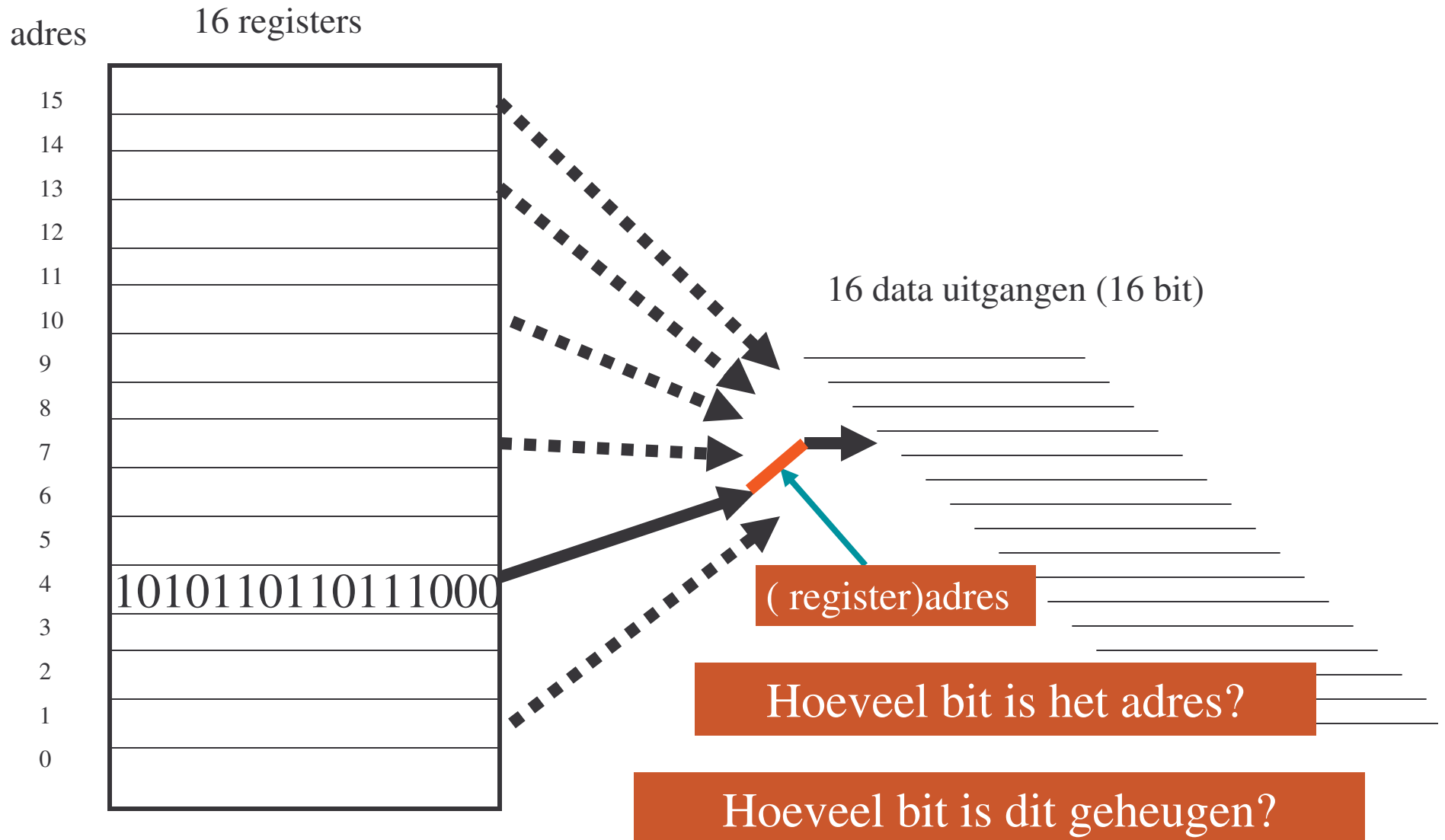


S_1	S_0	Operator	Functie
0	0	+ (plus)	$Y = A + B$
0	1	- (min)	$Y = A - B$
1	0	& (bitwise AND)	$Y = A \& B$
1	1	(geen operator)	$Y = B$

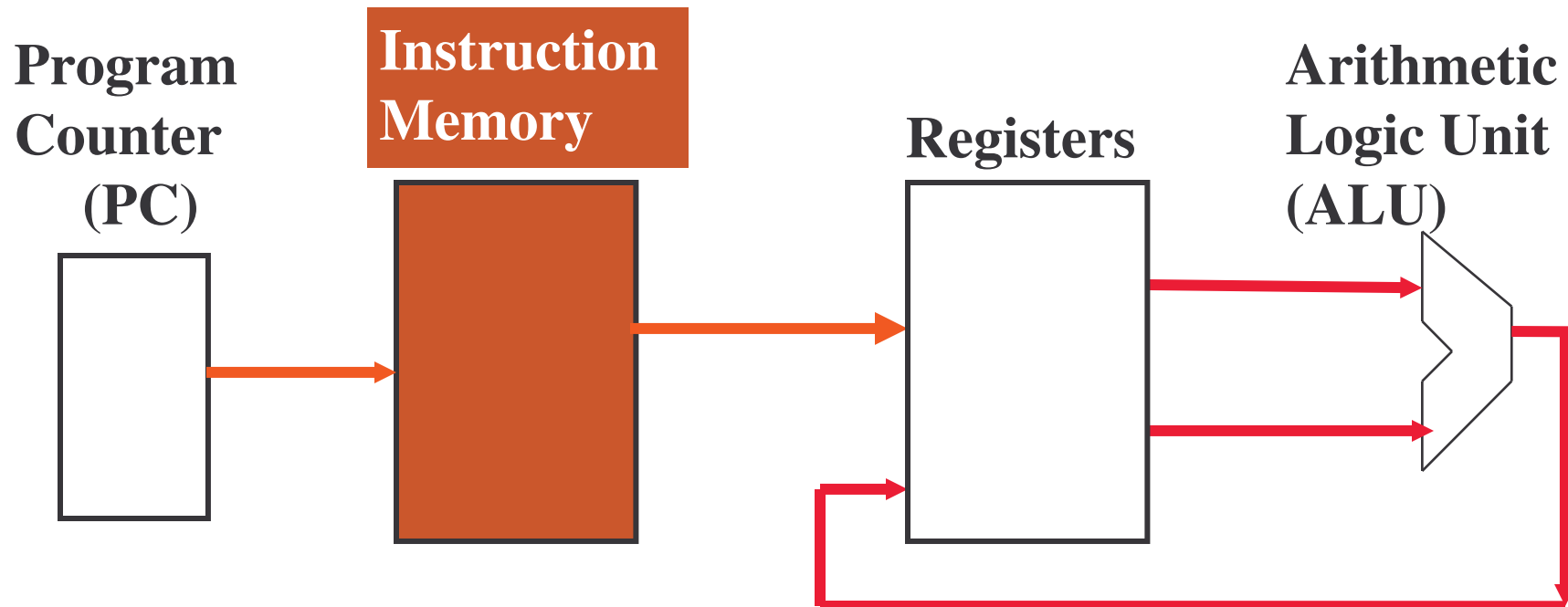
Registers



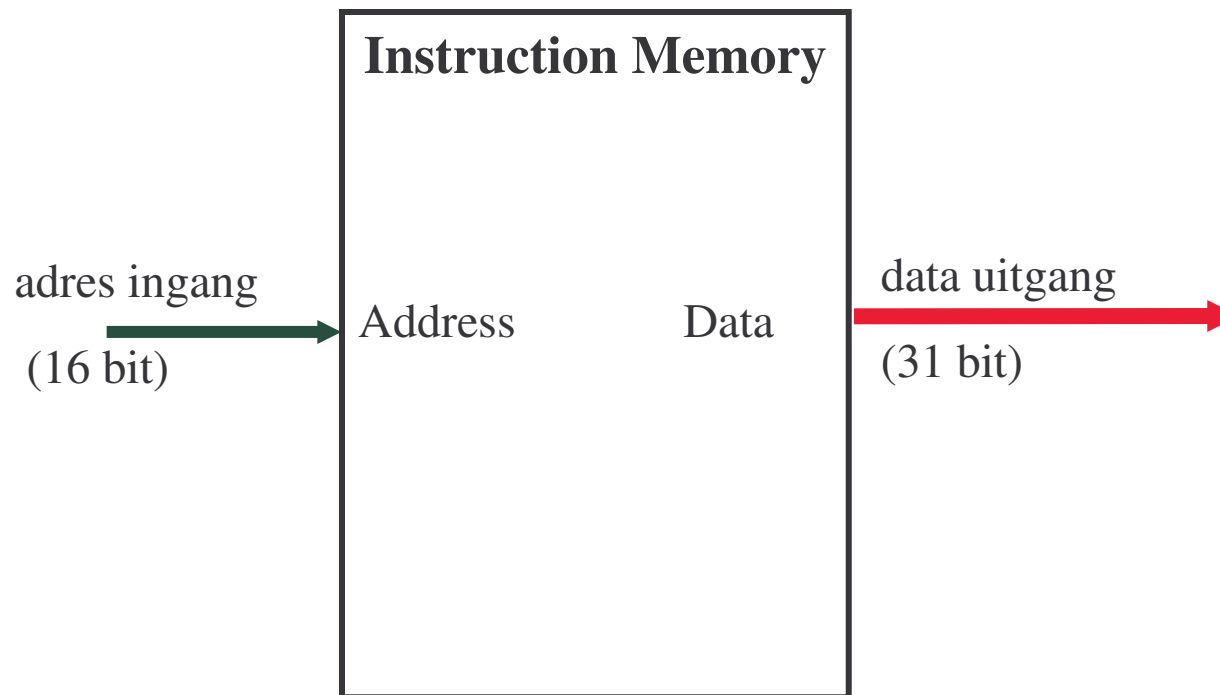
Registers



Instruction Memory

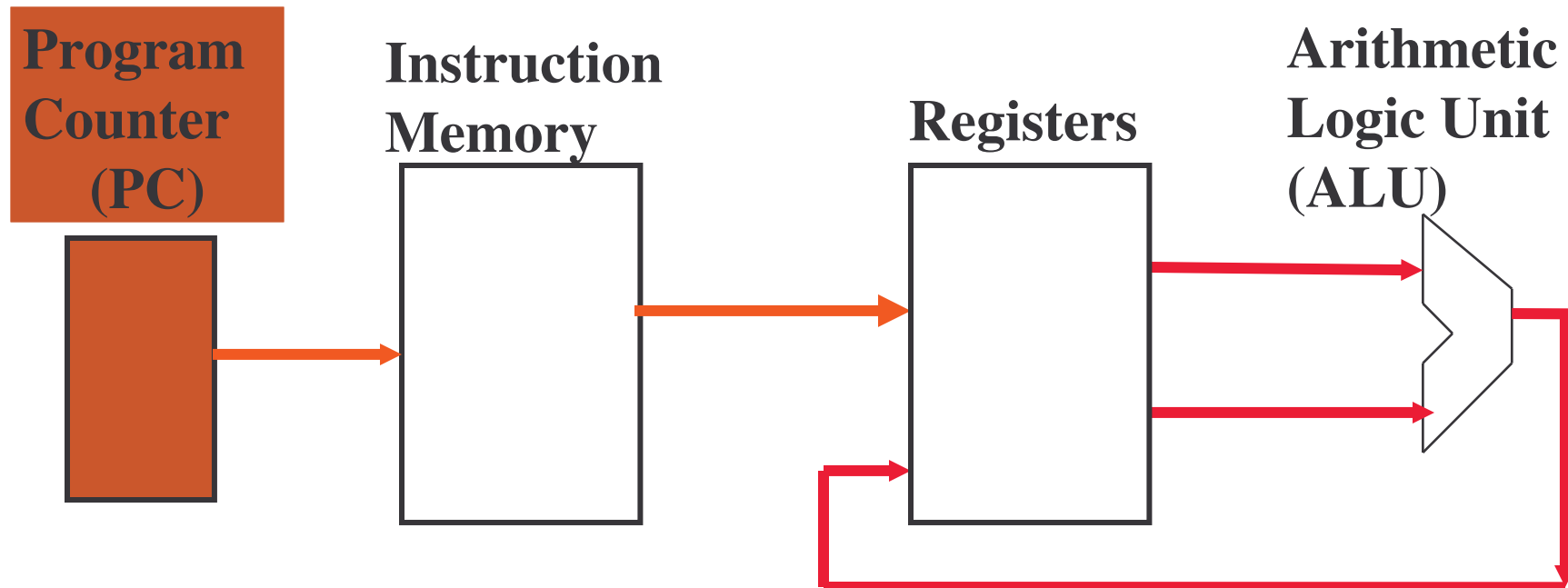


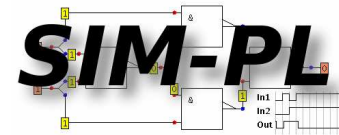
Instruction Memory



Aantal bits in dit geheugen: $2^{16} \times 31 = 65.536 * 31 = 2.031.616$

Program Counter

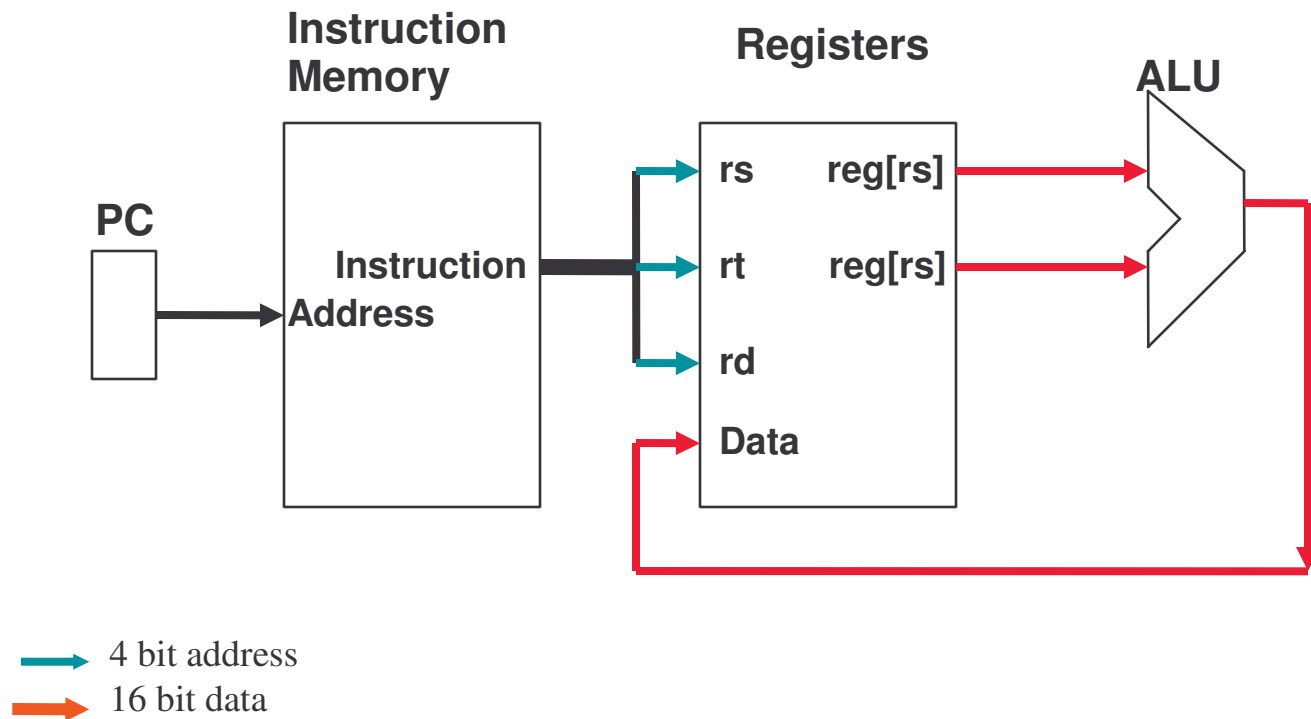




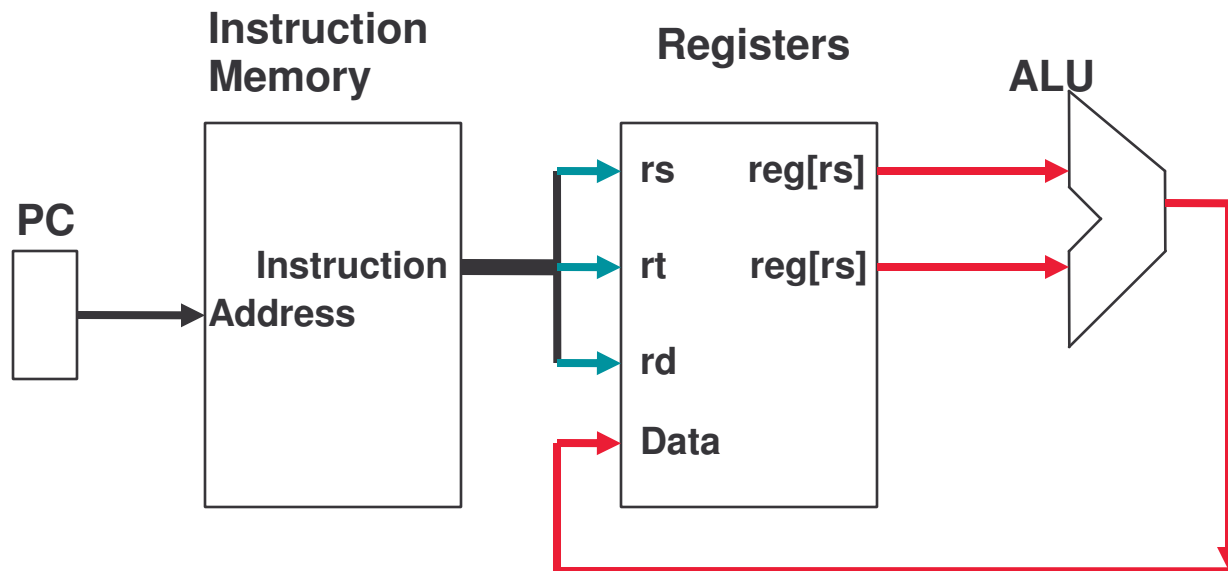
Instructies:

- Rekenkundige en logische instructies
 - ADD
 - SUB
 - AND
 - MOVE
- Immediate instructies (Instructies met één constant getal)
 - ADDI
 - SUBI
 - ANDI
 - LOAD

Datapad van een rekenkundige/logische instructie: bijv. ADD



Assembly Language

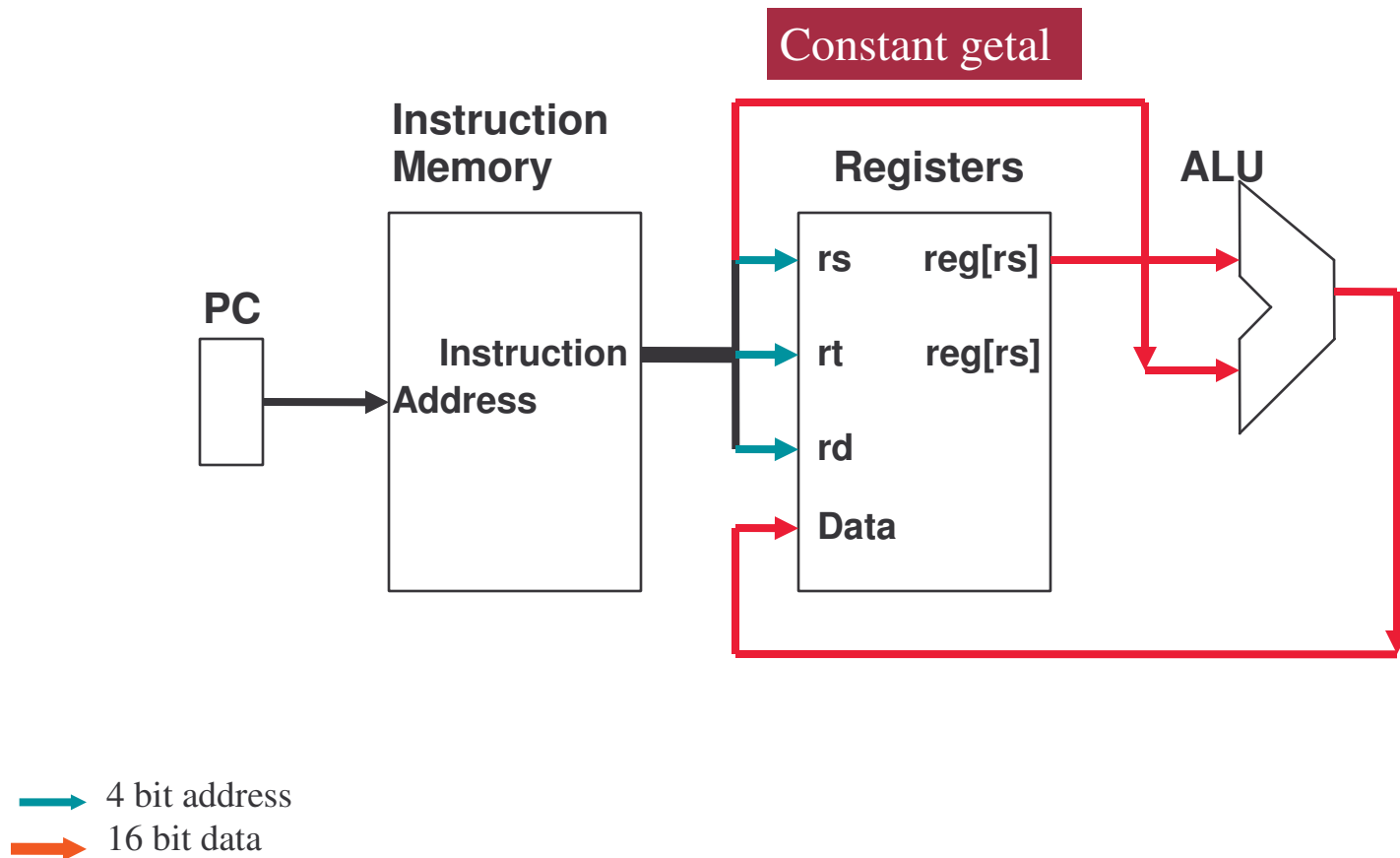


Syntax : ADD rd, rs, rt

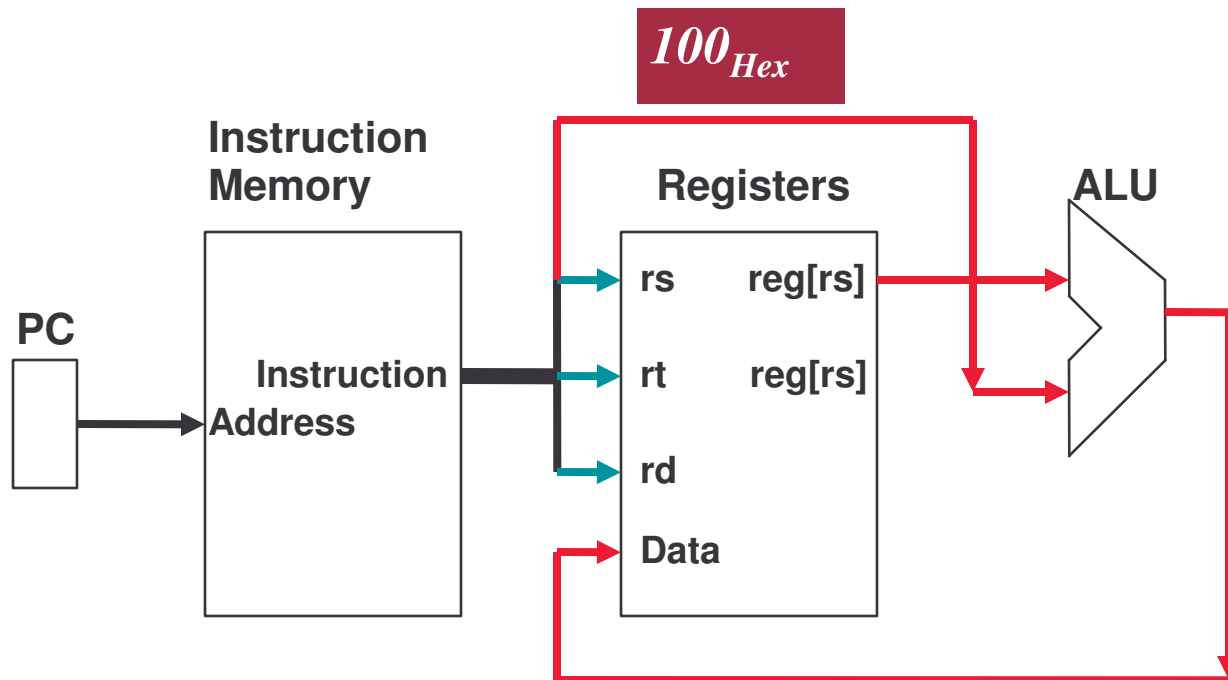
Voorbeeld: ADD \$7, \$5, \$6

Betekenis: register 7 = register 5 + register 6

Datapad van een instructie met een constant getal



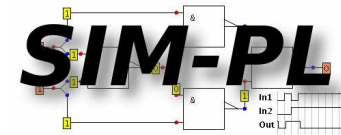
Assembly Language:



Syntax: *ADDI rd, rs, getal*

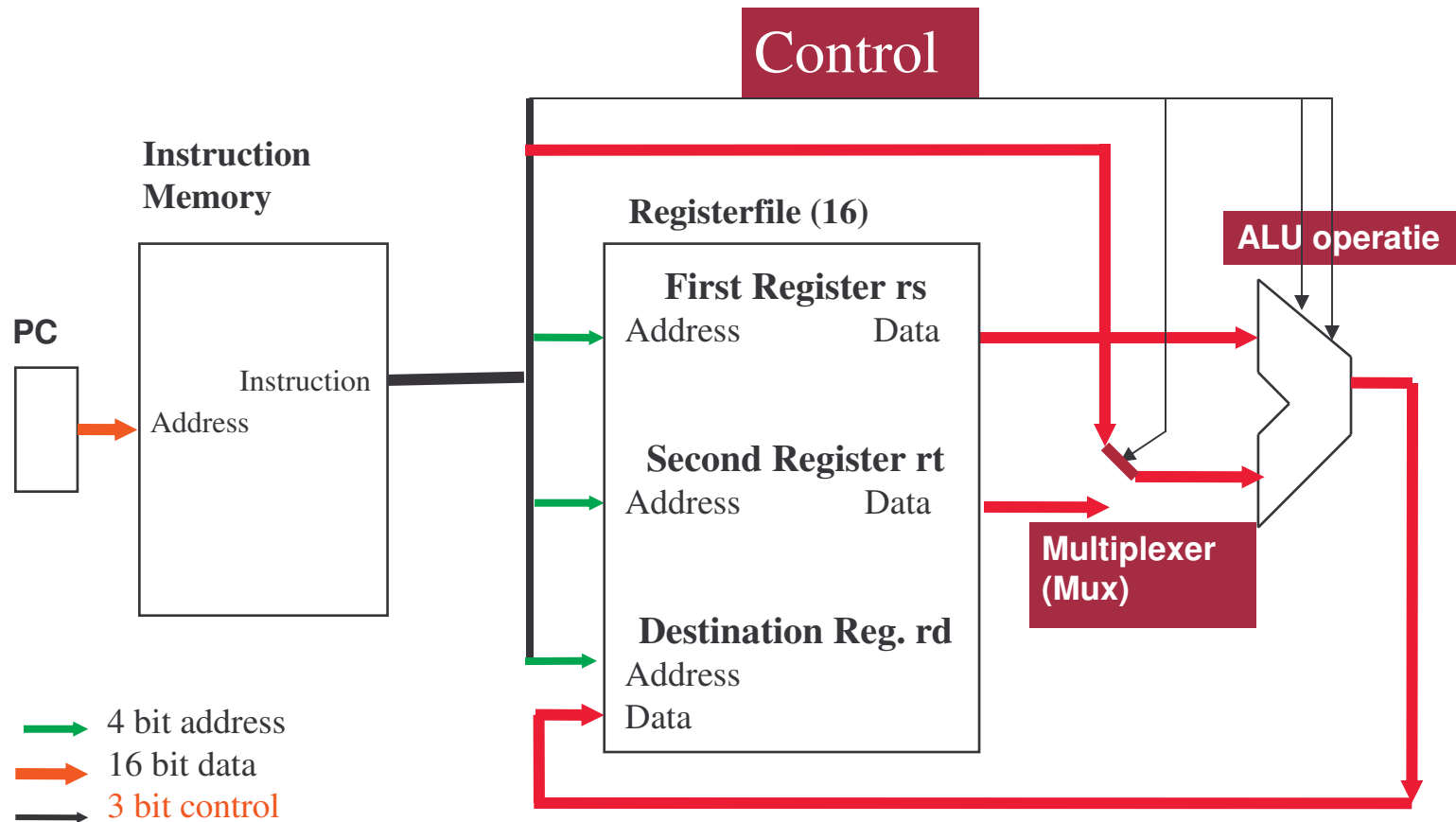
Voorbeeld: *ADDI \$7, \$5, 0x100*

Betekenis: *register 7 = register 5 + 100_{Hex}*

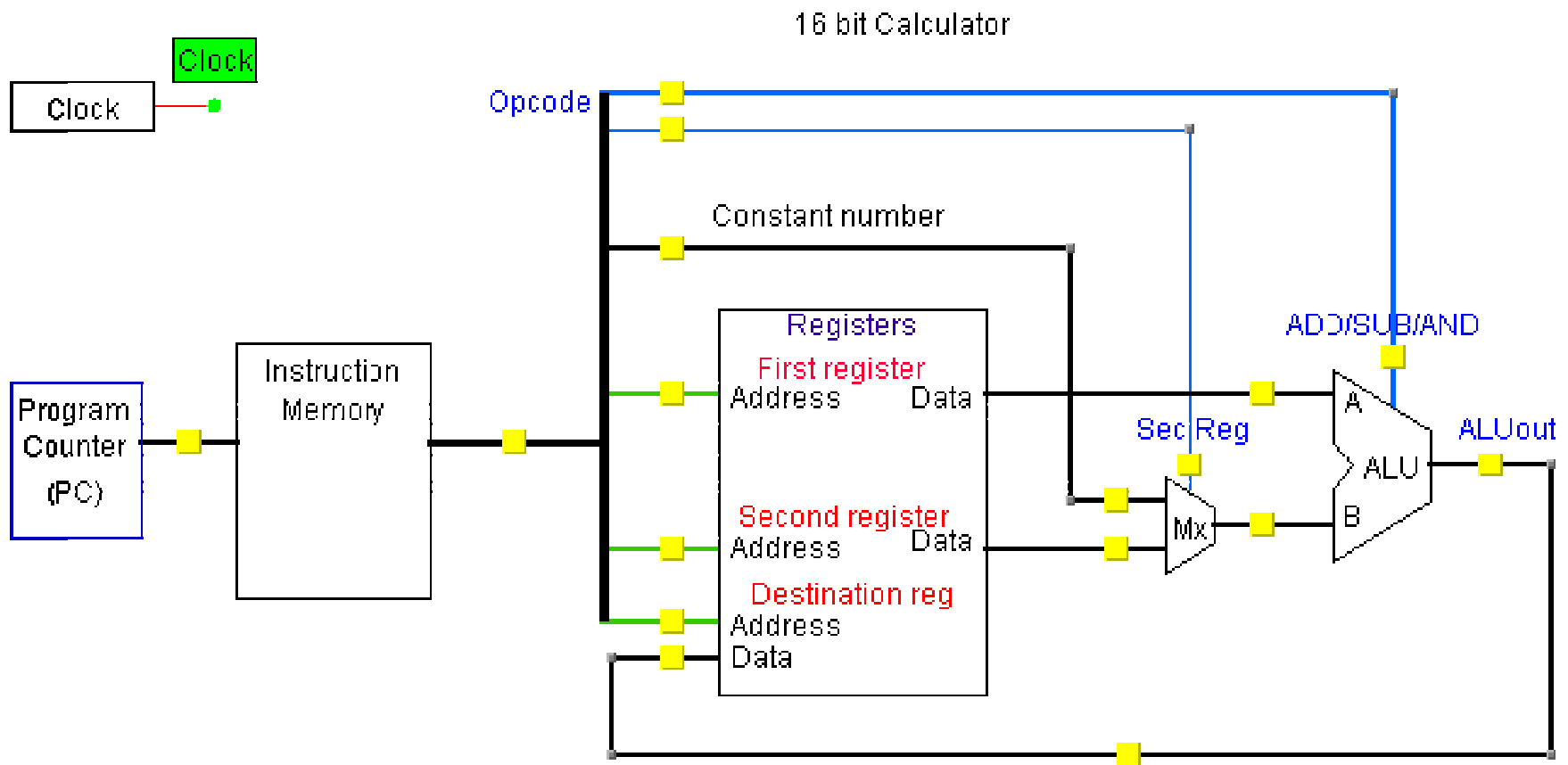


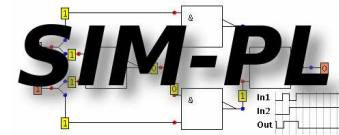
Voer opdracht 1 en opdracht 2 uit op bladz. 6 en
7

“Control” van de rekenmachine



Demonstratie 16 bit calculator

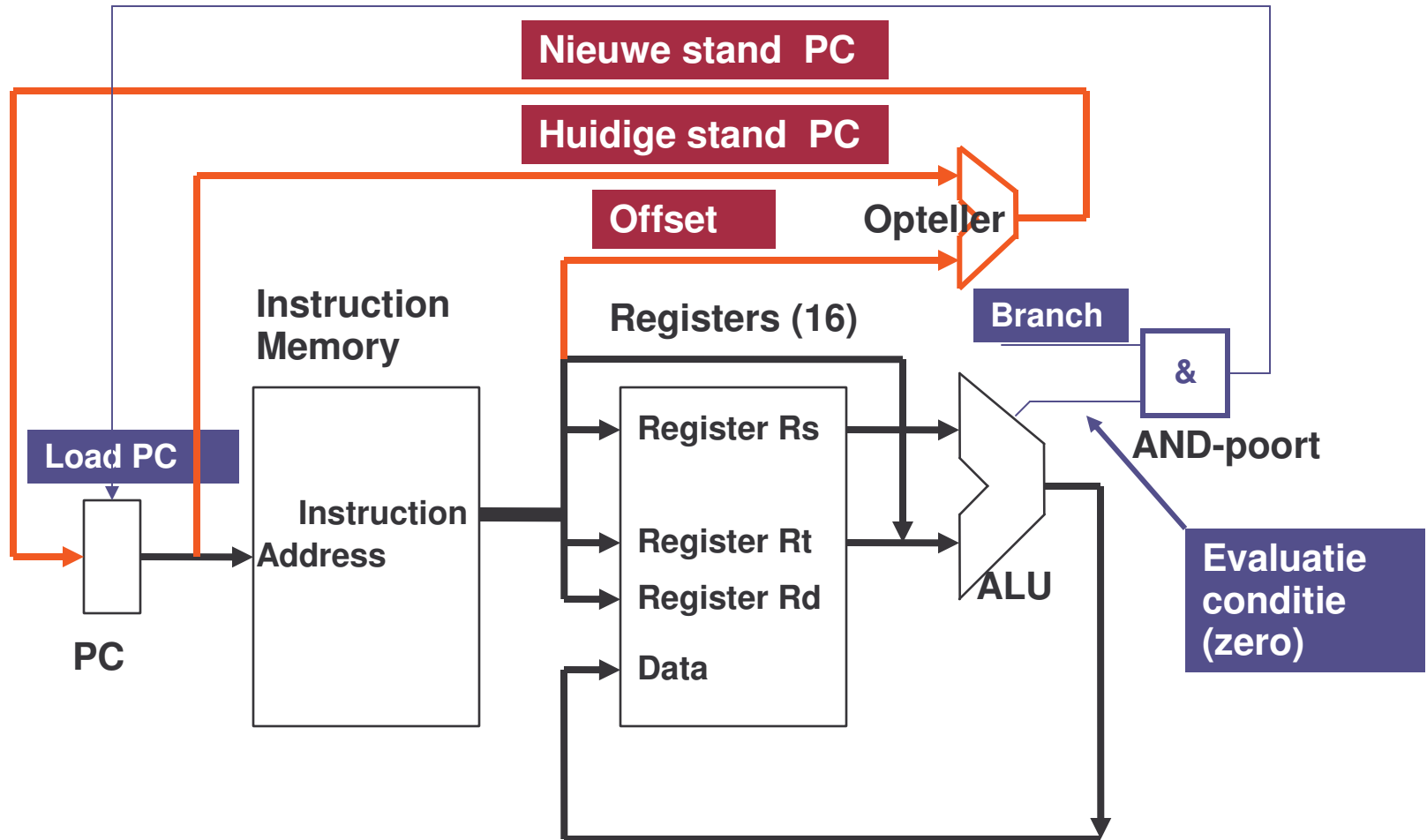


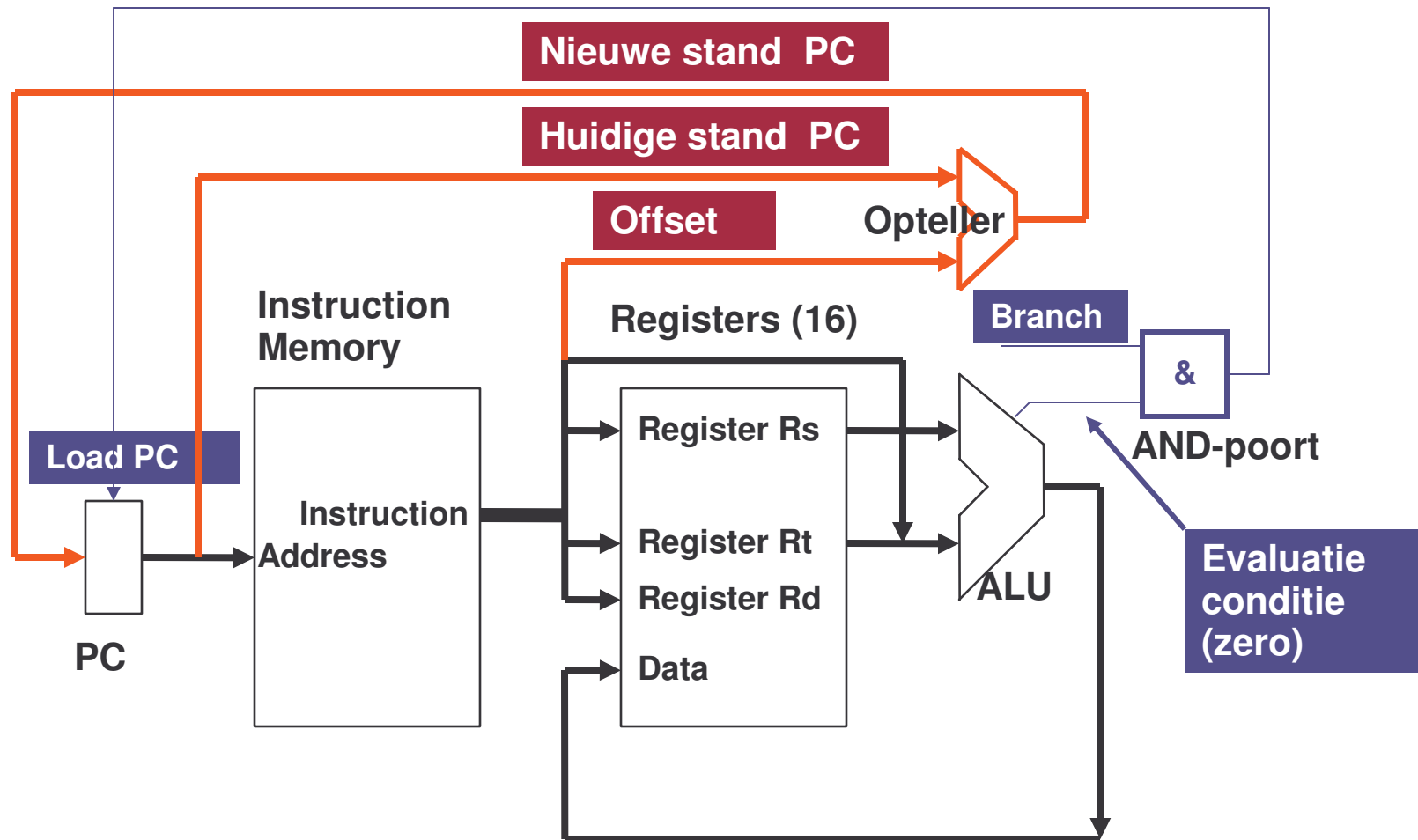


Branch instructies:

- Rekenkundige en logische instructies
 - ADD
 - SUB
 - AND
 - MOVE
- Immediate instructies (Instructies met één constant getal)
 - ADDI
 - SUBI
 - ANDI
 - LOAD
- Branch instructies
 - BZ (Branch if zero)
 - BEQ (Branch if equal)
 - BRA (Branch always)

Uitbreiding loopjes: opteller en AND-poort



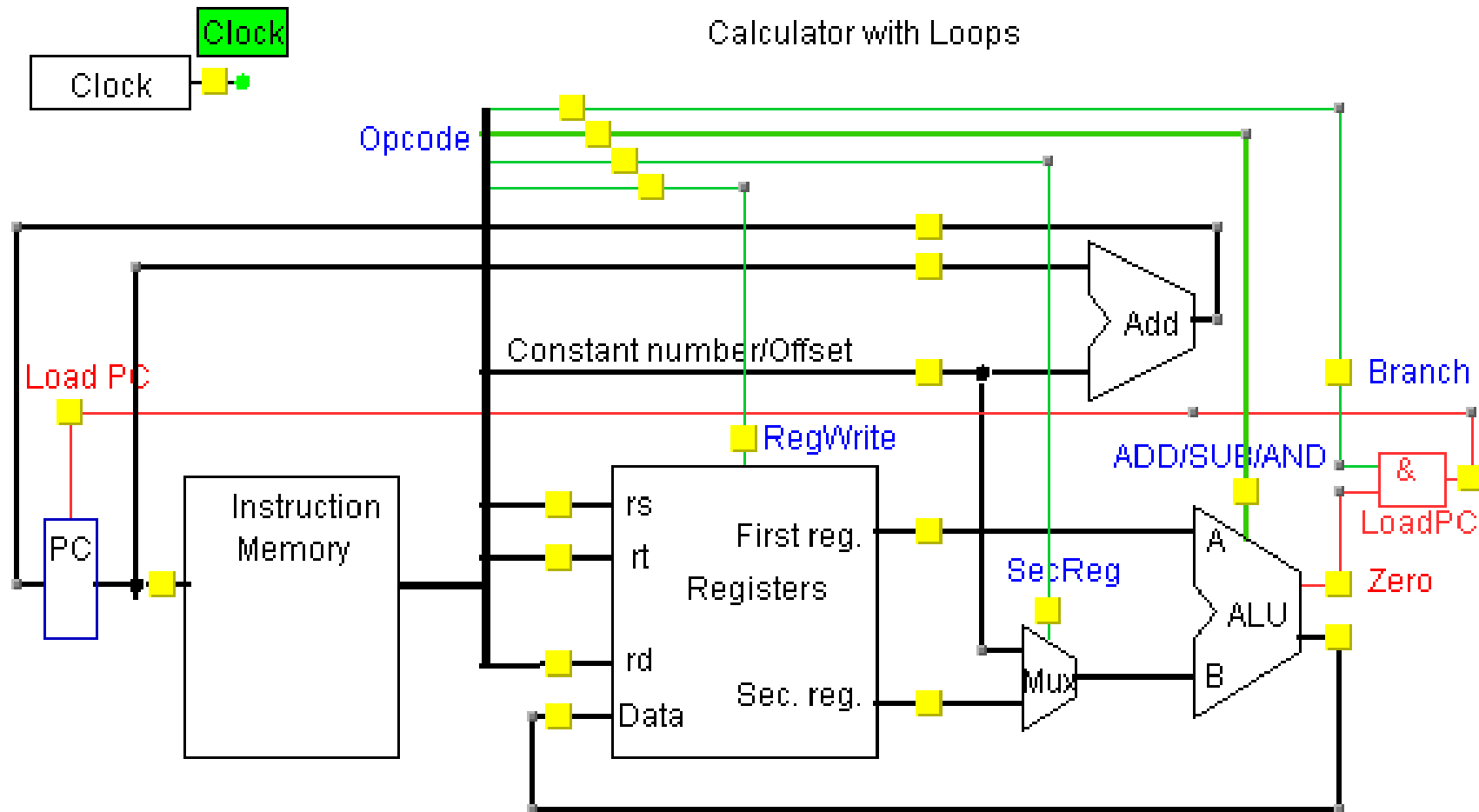


*Syntax : **BEQ** rs, rt, label*

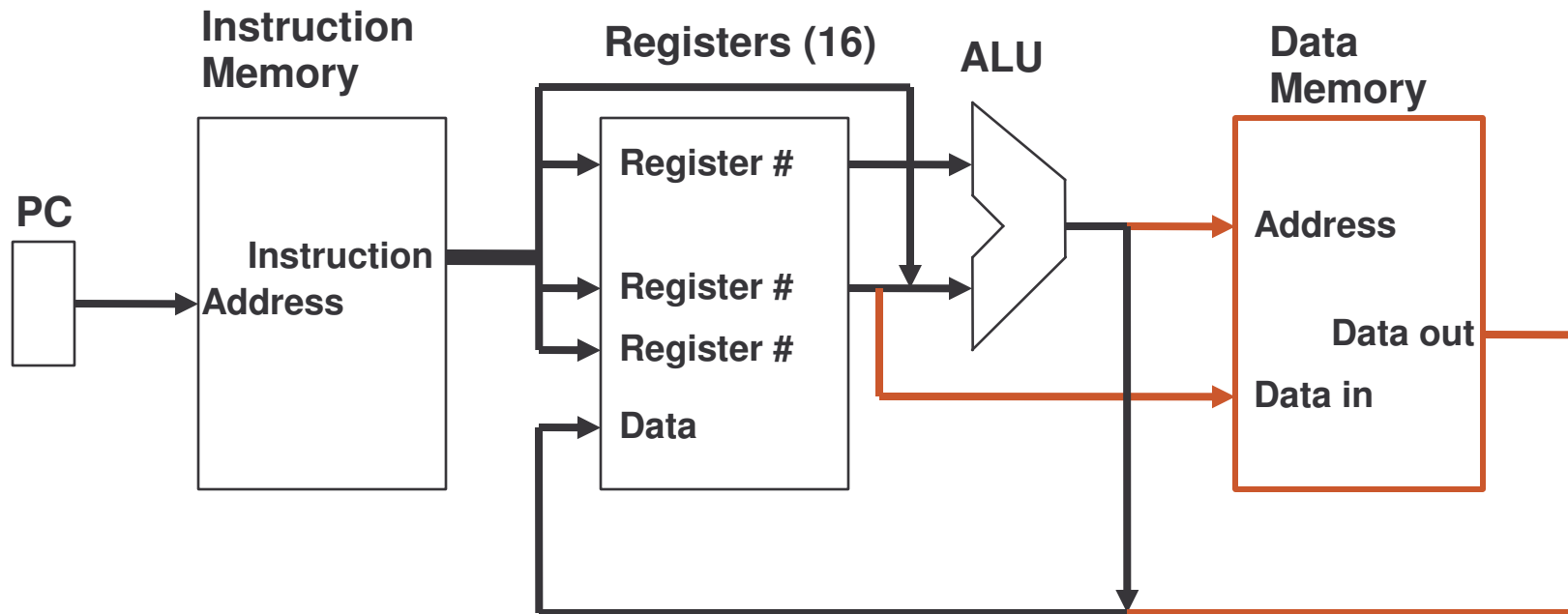
*Voorbeeld: **BEQ** \$7, \$5, loop*

*Betekenis: **if** (register 7 = register 5) goto loop*

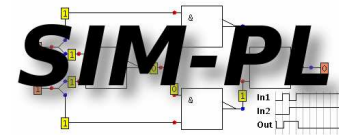
Demo Calculator with loops



Uitbreiding met Data Memory

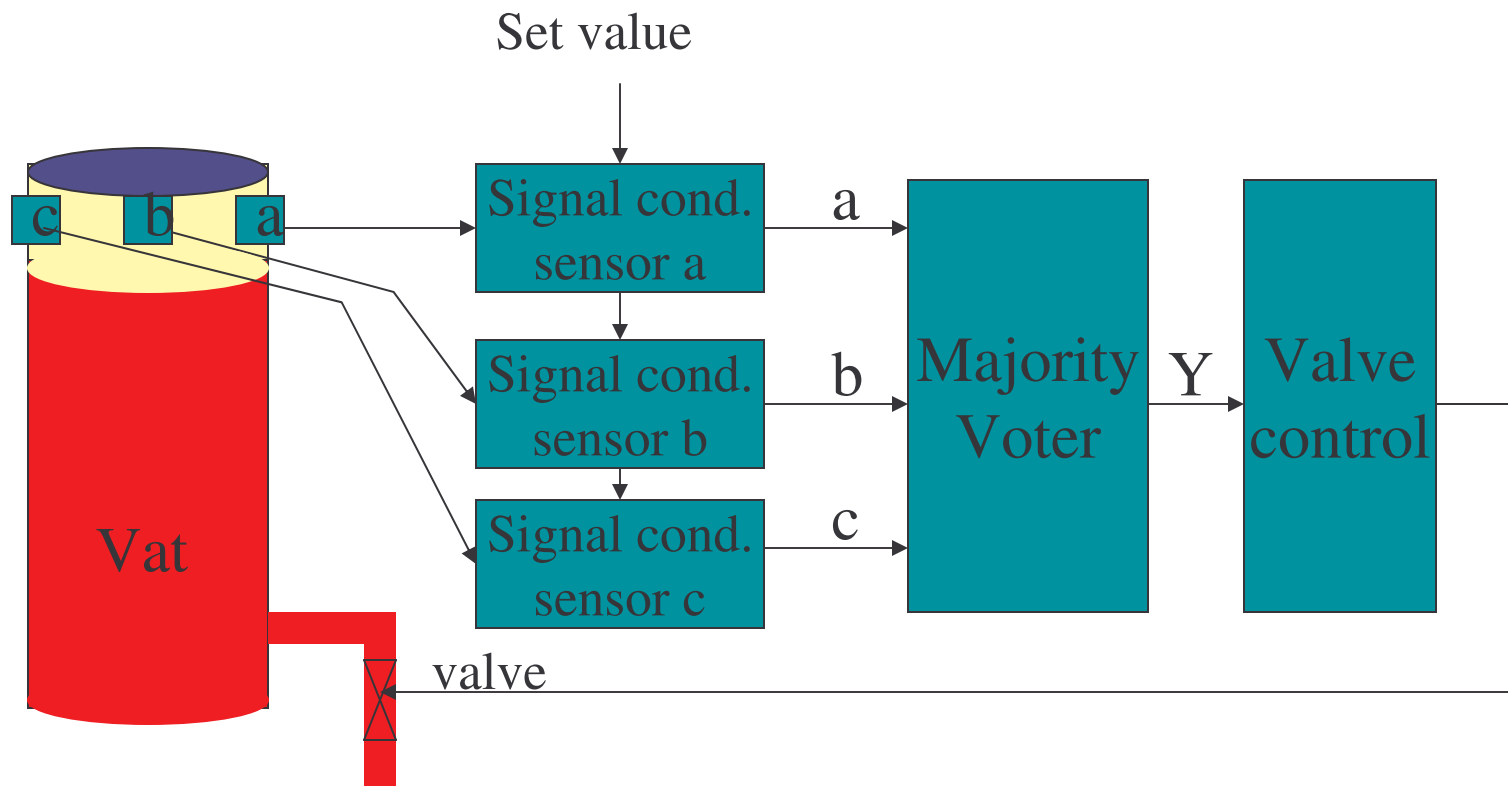


Bij een **Harvard architectuur** is het geheugen gescheiden in twee delen: instructiegeheugen en datageheugen(zie boekje)



Ontwerptool voor schakelingen

Voorbeeld poortschakeling: “Majority Voter”



■ redundant system

Waarheidstabel “Majority Voter”

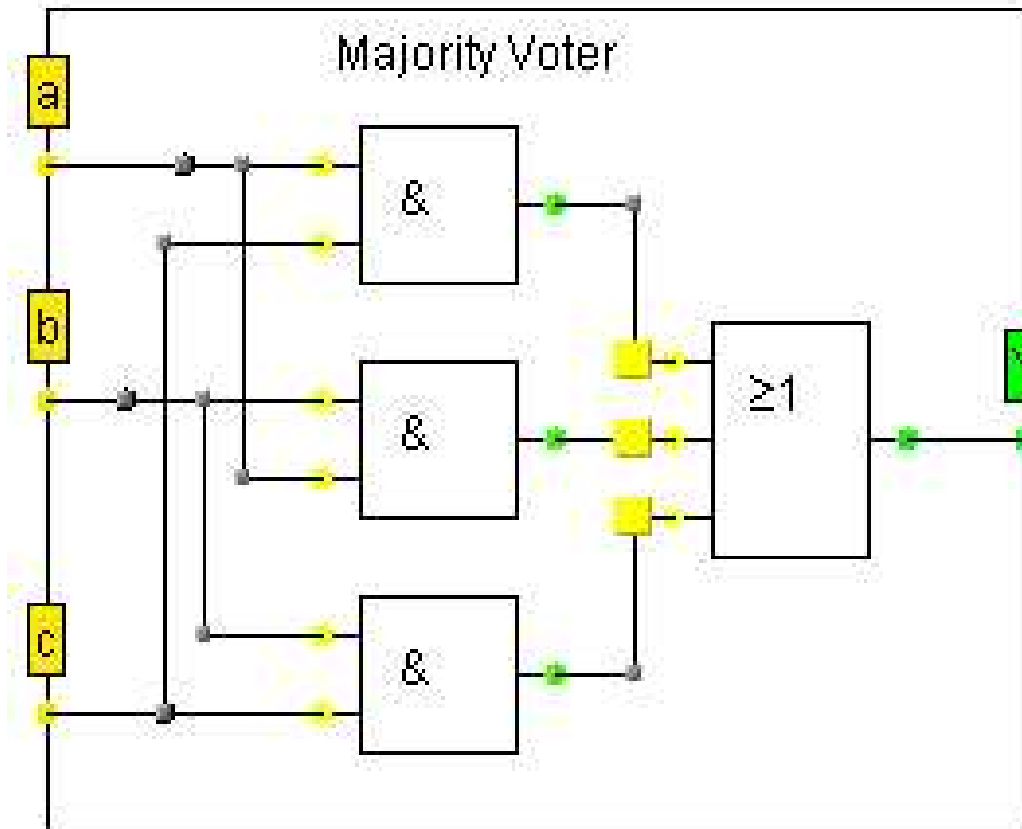
a	b	c	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Formule: $Y = (a \text{ AND } b) \text{ OR } (a \text{ AND } c) \text{ OR } (b \text{ AND } c)$

Benodigde componenten:

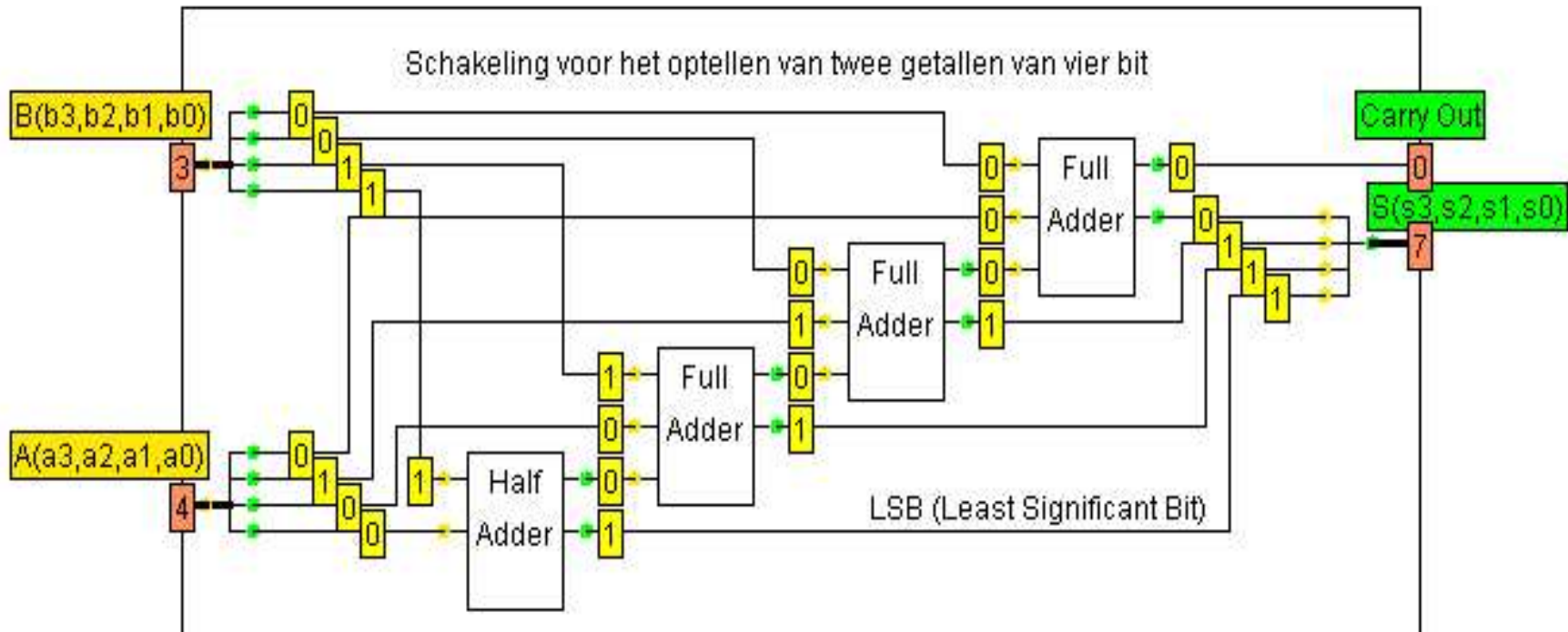
- Drie AND-poorten met twee ingangen
- Eén OR-poort met drie ingangen

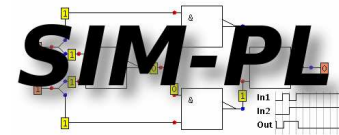
Schema “Majority Voter”



$$Y = (a \text{ AND } b) \text{ OR } (a \text{ AND } c) \text{ OR } (b \text{ AND } c)$$

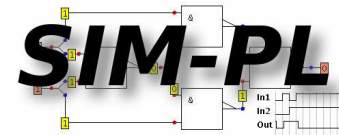
Ander voorbeeld: 4 bits opteller





SIM-PL in het VO

- November 2005: Materiaal getest met leerlingen van Ronald Leijtens (Damstede)
 - Geschikt voor VWO; HAVO?
- Januari 2006: SIM-PL 2.0 gereed
- Voorjaar 2006: Materiaal testen met diverse scholen
 - Zijn er onder jullie nog belangstellenden om te testen?
- Juli 2006: Lesboek en docenthandleiding voor zes tot acht lessen gereed
- Juli 2006: Aanbieden materiaal aan commissie urenuitbreiding 2007
- Website als “Community of practice”
<http://www.science.uva.nl/amstel/SIM-PL>



Vragen?

- Software
 - Executer (Simulator)
 - Editor
- Bibliotheek modellen
 - Van poort tot processor
- Lesmateriaal
 - Presentaties
 - Practica

<http://www.science.uva.nl/amstel/SIM-PL>