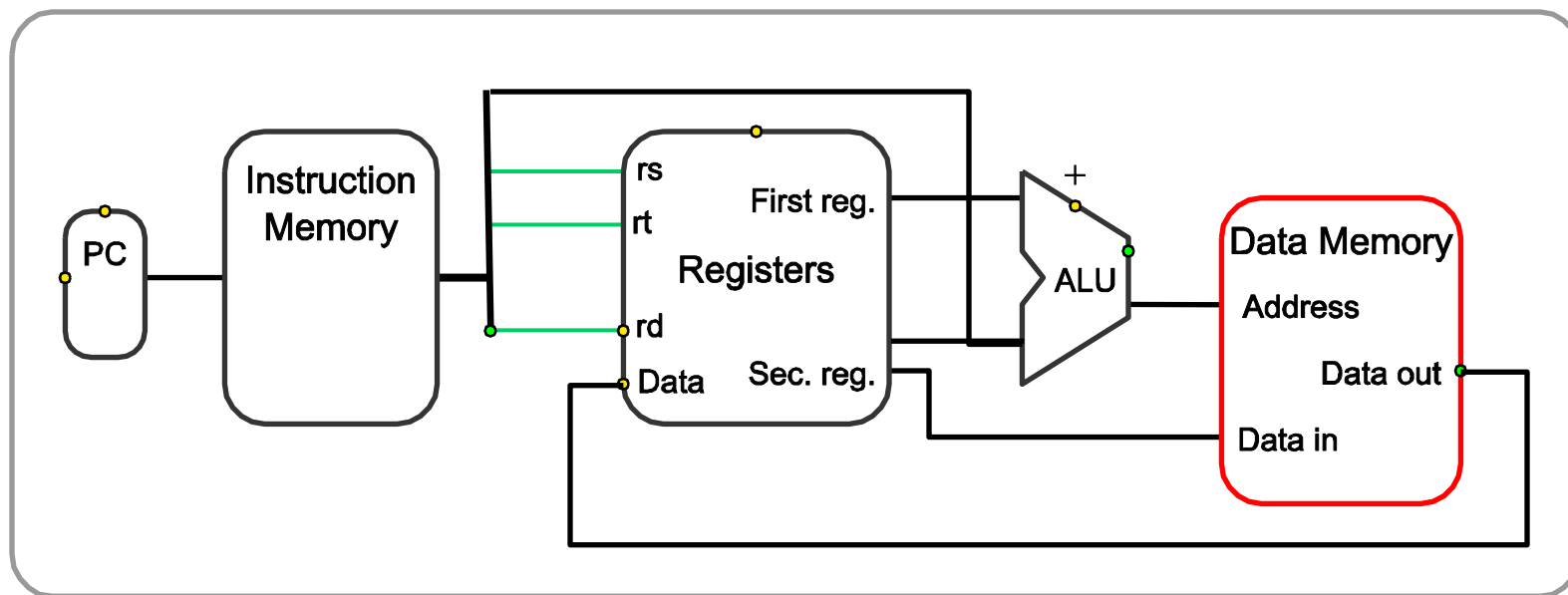




# De Harvard Machine

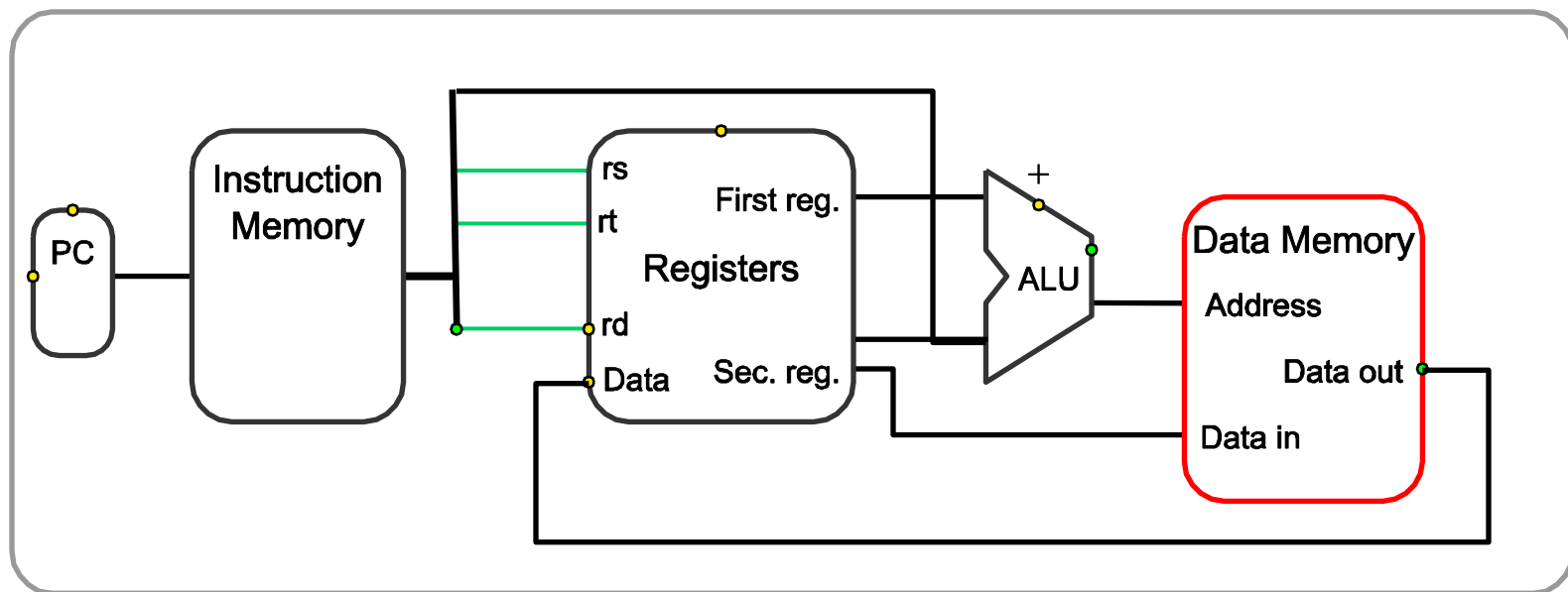
- Van rekenmachine met “loopjes” naar processor

# Extra component: Data Memory



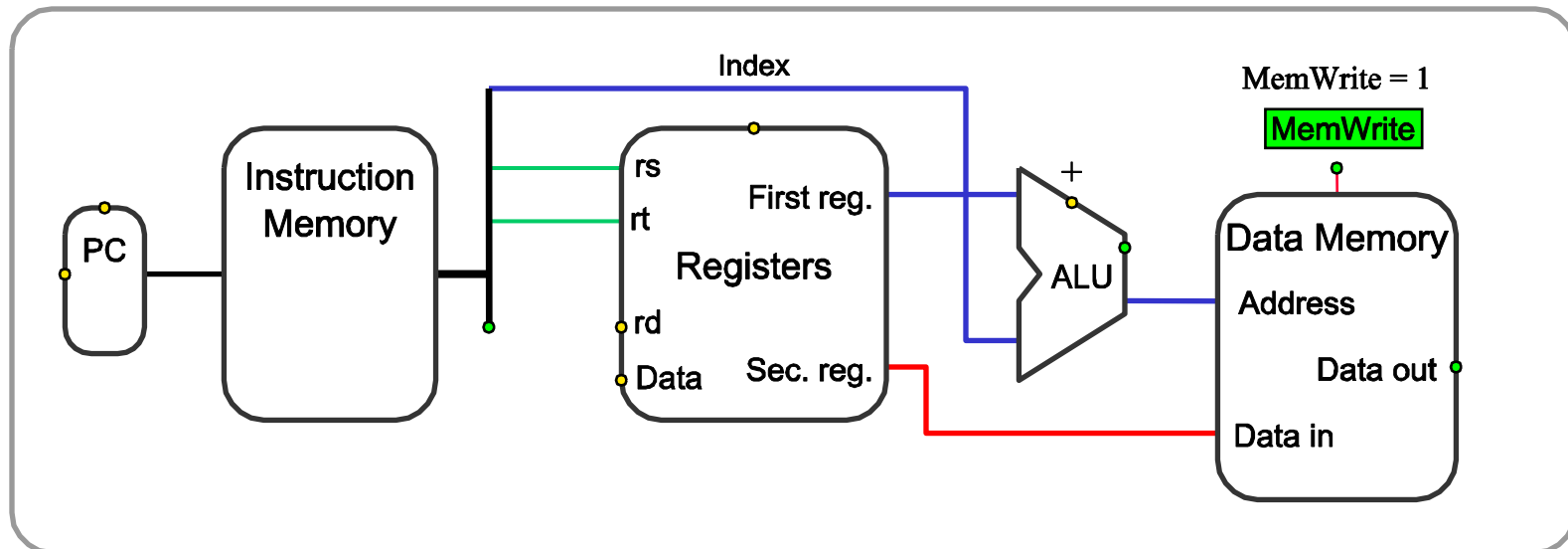
Bij een Harvard architectuur is het geheugen gescheiden in twee delen: Het instructiegeheugen en het datageheugen

# Extra instructies: Store Word & Load Word



Slechts twee instructies communiceren met het datageheugen:  
Load/Store Machine

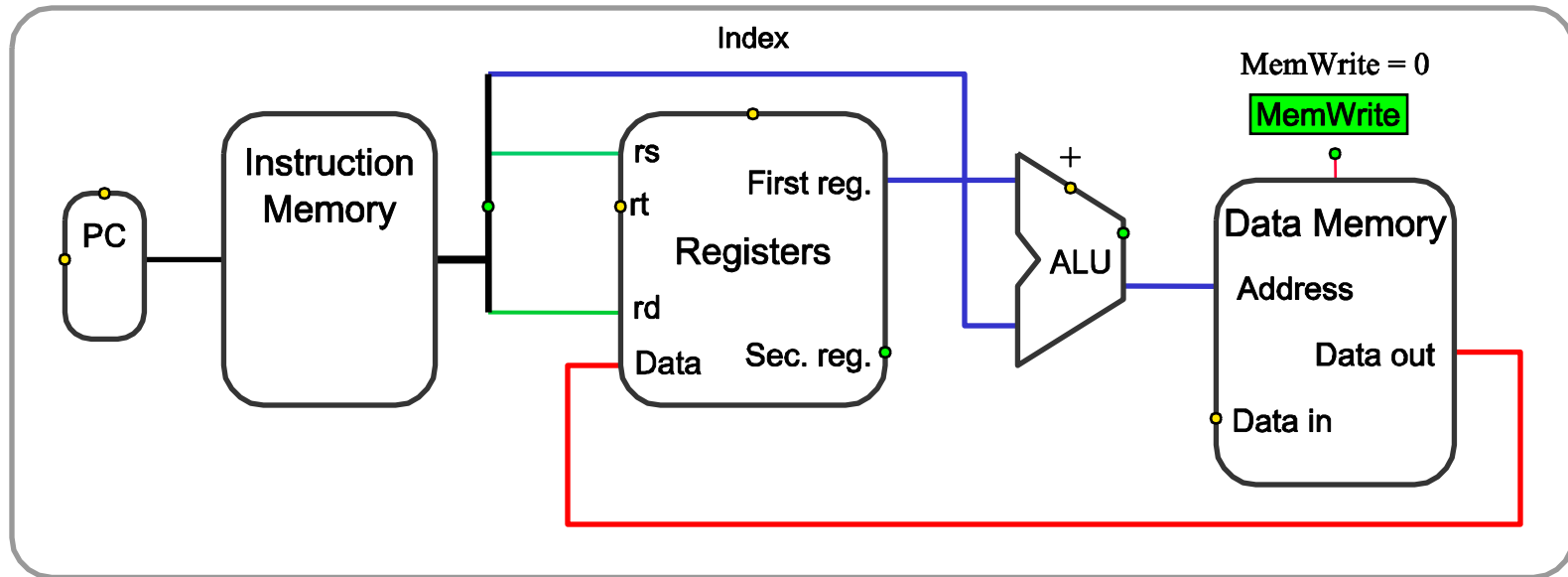
# Datapad Store Word-instructie



Store Word (SW): Second register → Data Memory

SW rt, index , rs  
 SW \$0, 0x10, \$1  
 $\text{Memory}[\text{R1} + 10_{\text{HEX}}] = \text{R0}$

# Datapad Load Word-instructie



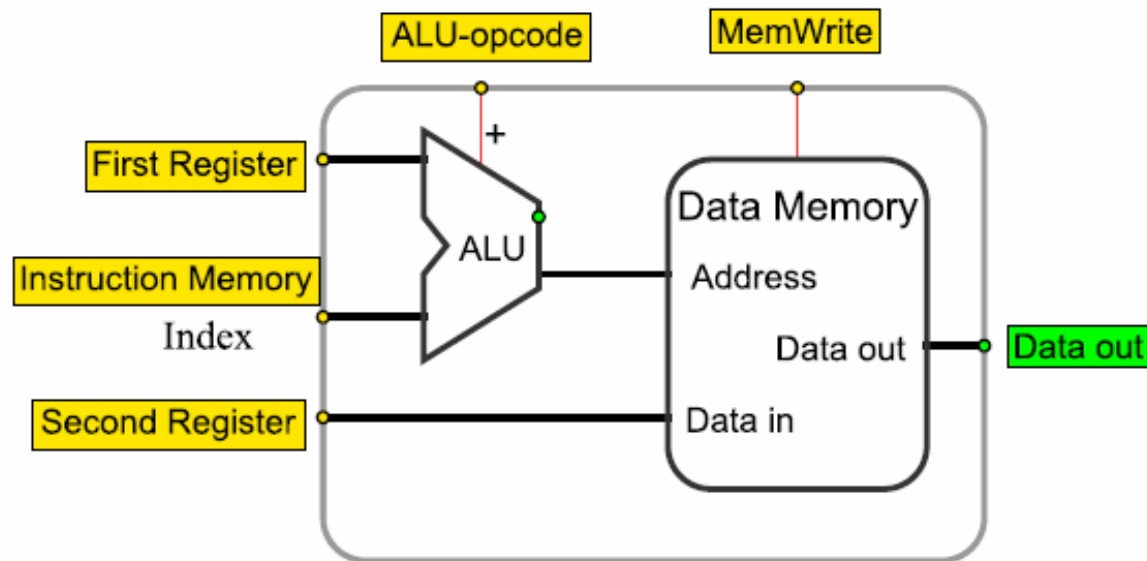
Load Word (LW): Destination register  $\leftarrow$  Data Memory

LW *rd*, *index* , *rs*

LW \$1, 0x10, \$2

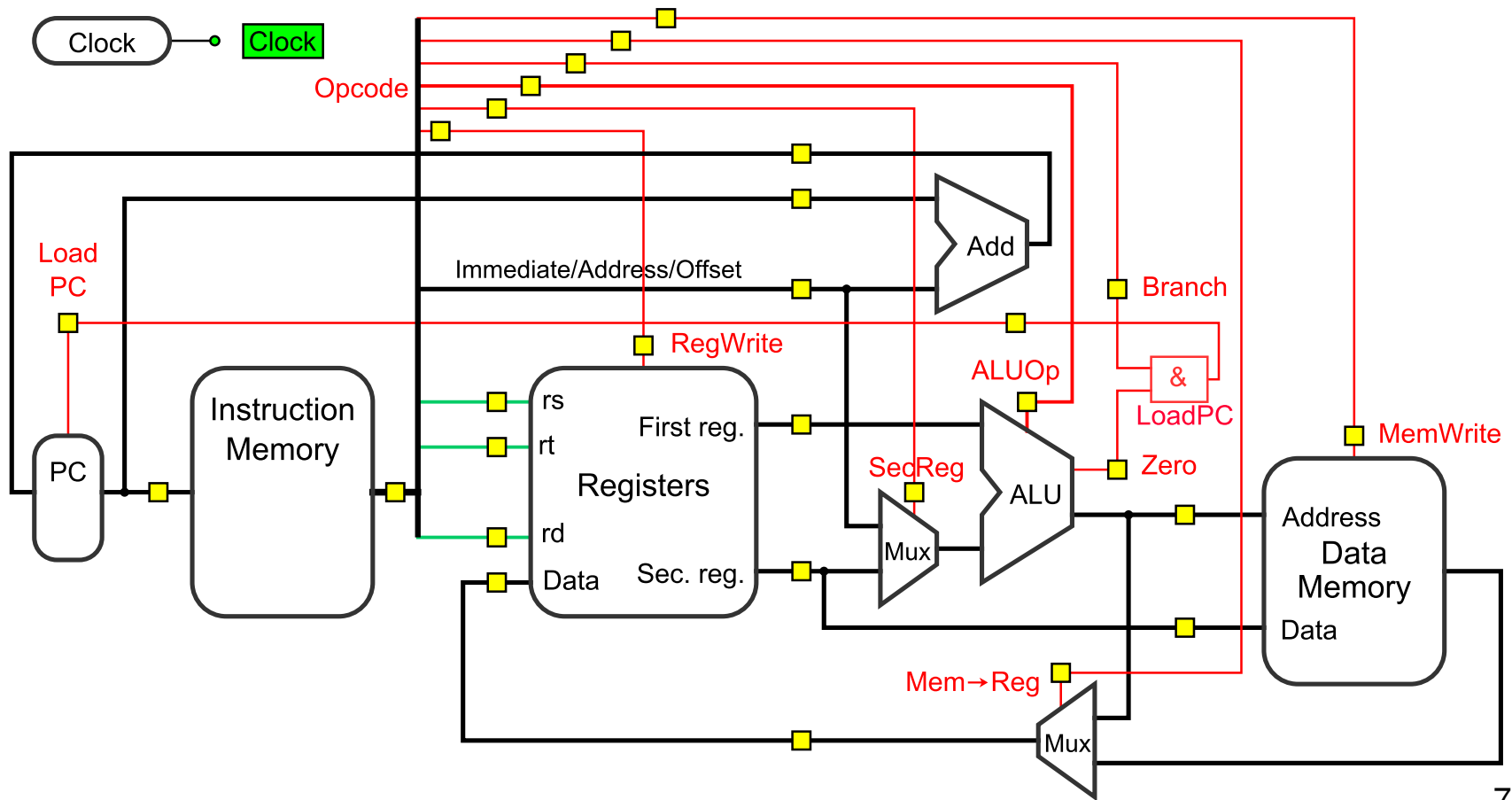
$R0 = \text{Memory}[R1 + 10_{\text{HEX}}]$

# ALU & Data Memory

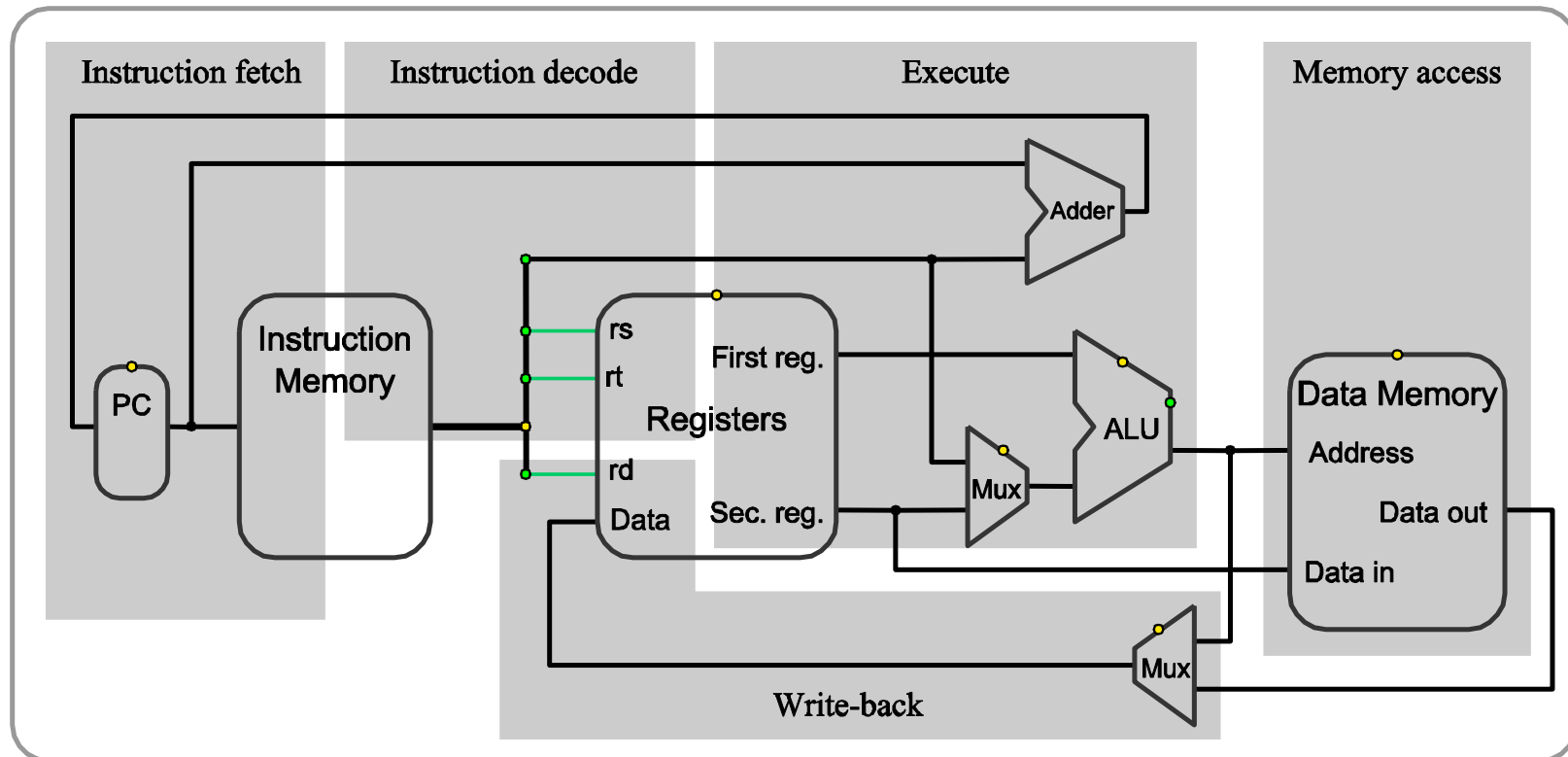


# De Harvard machine

16 bit Harvard Architecture

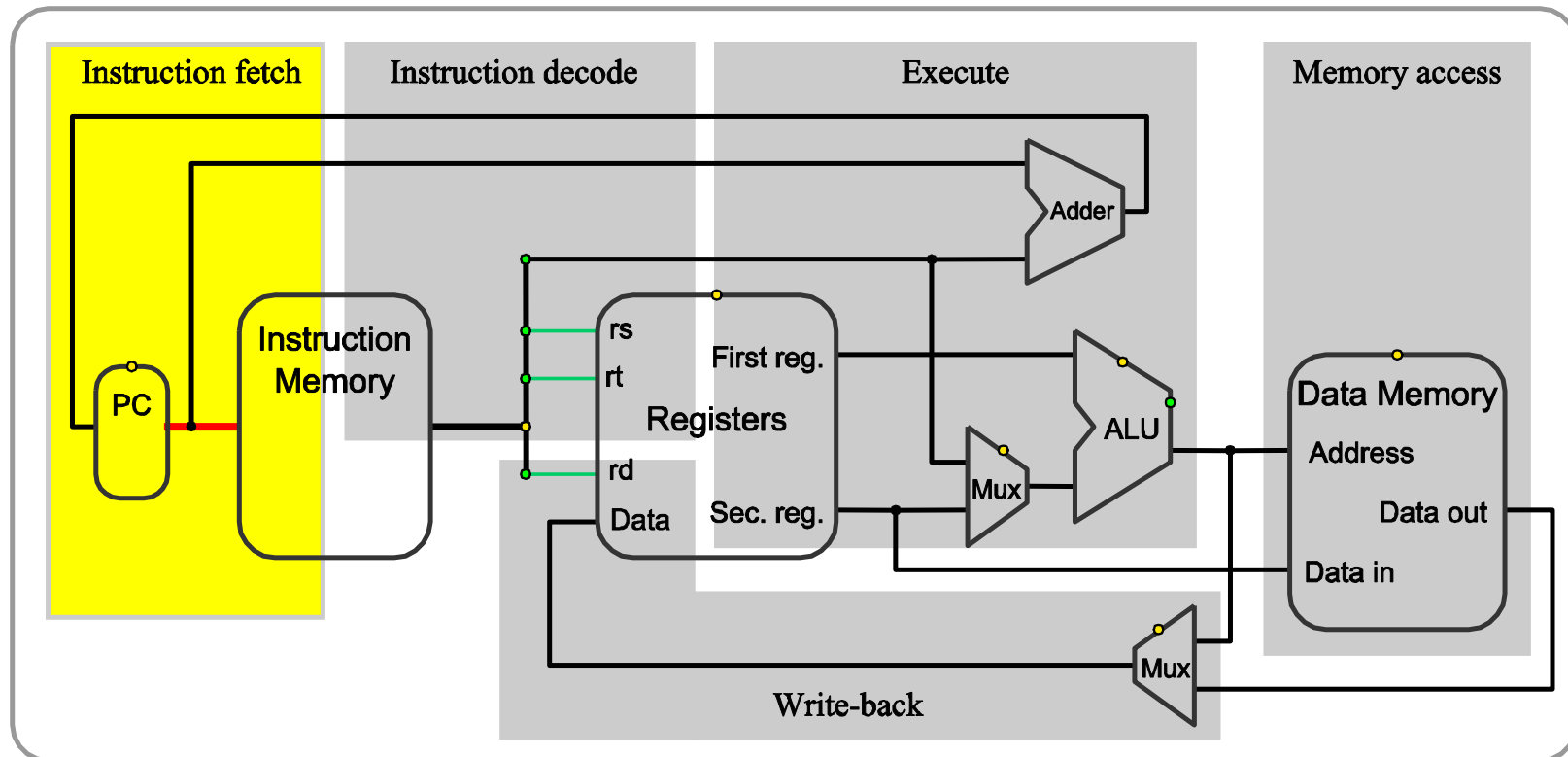


# De vijf fases van een instructie

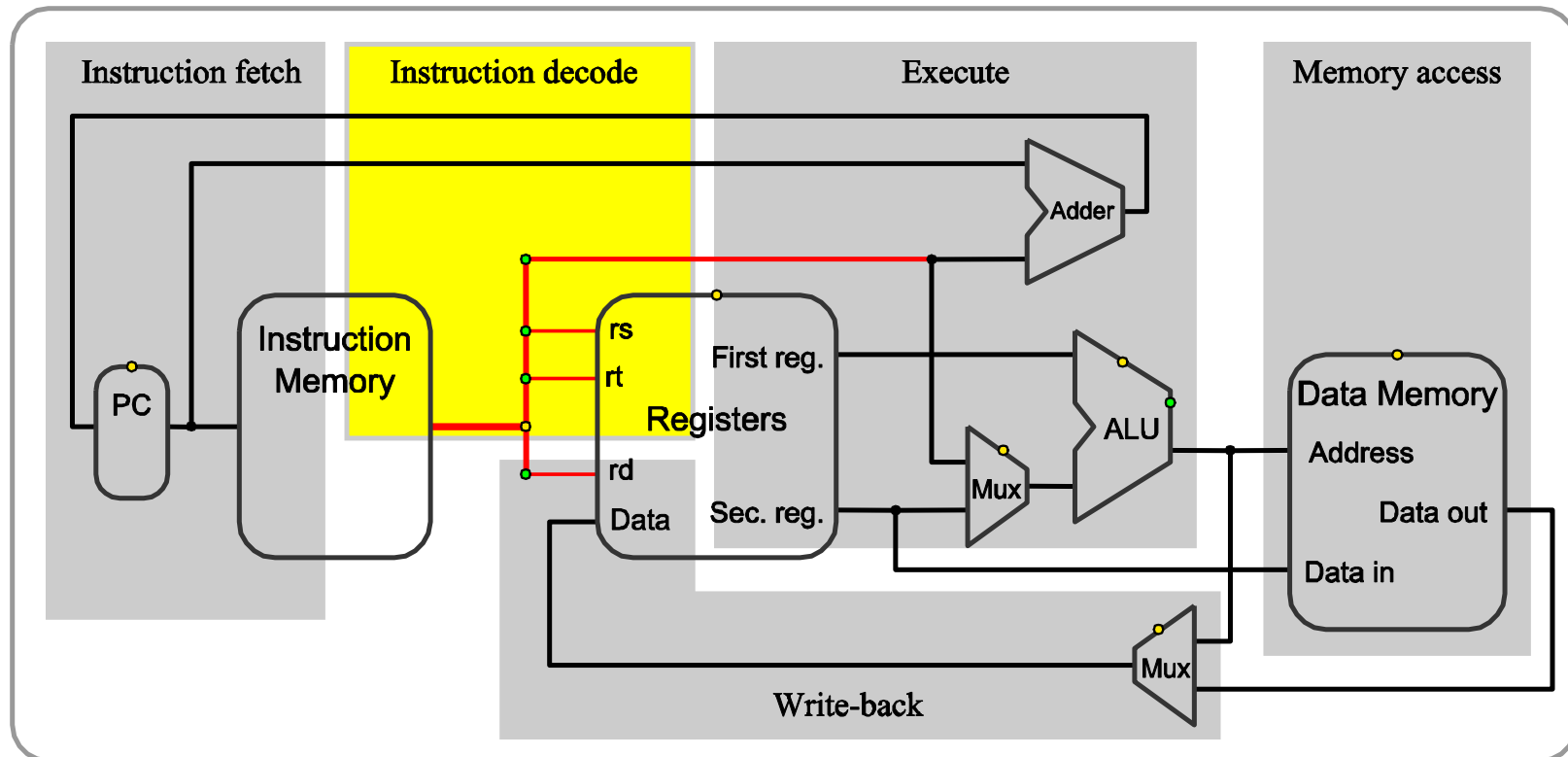




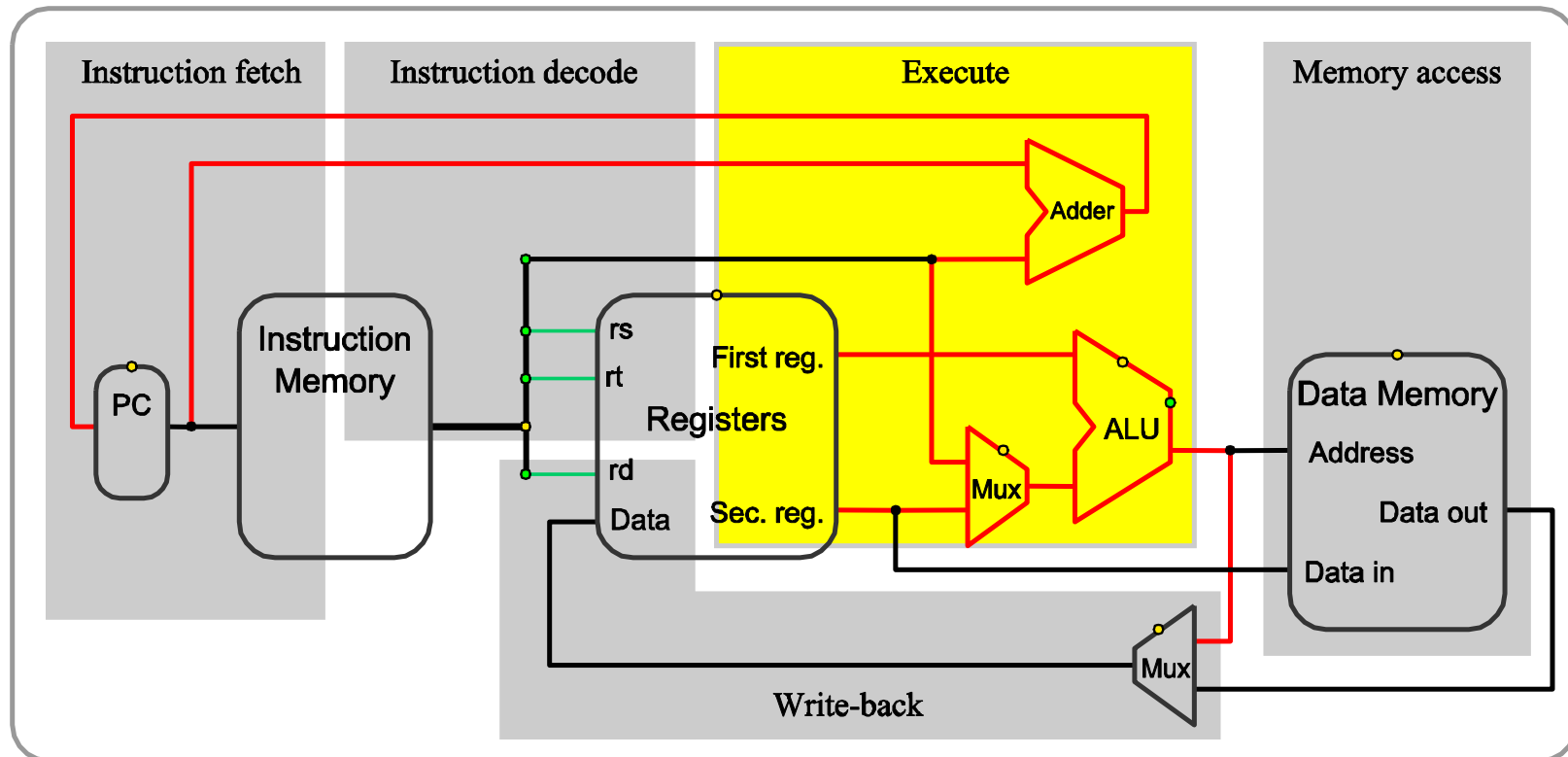
# De vijf fases van een instructie



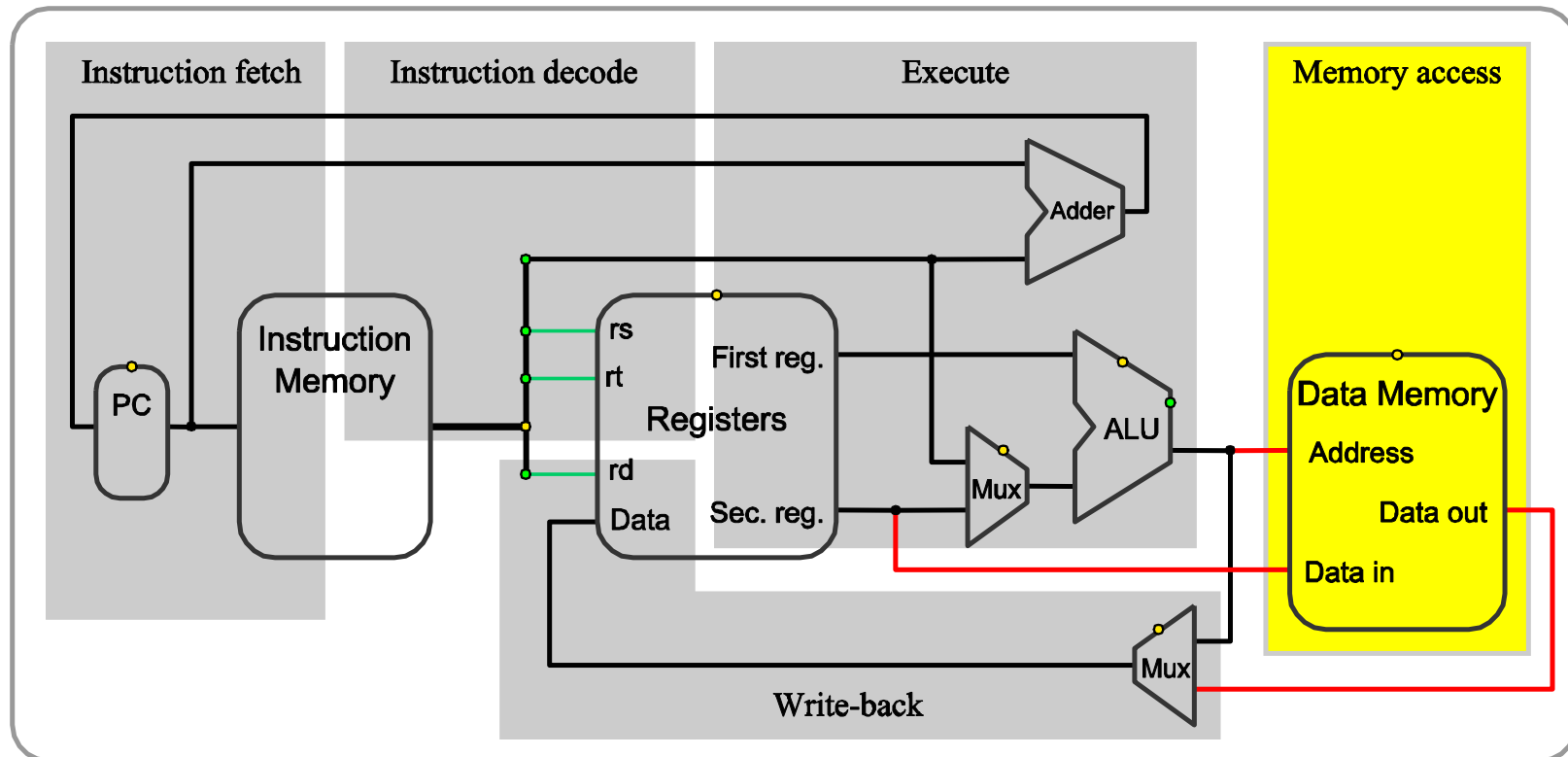
# De vijf fases van een instructie



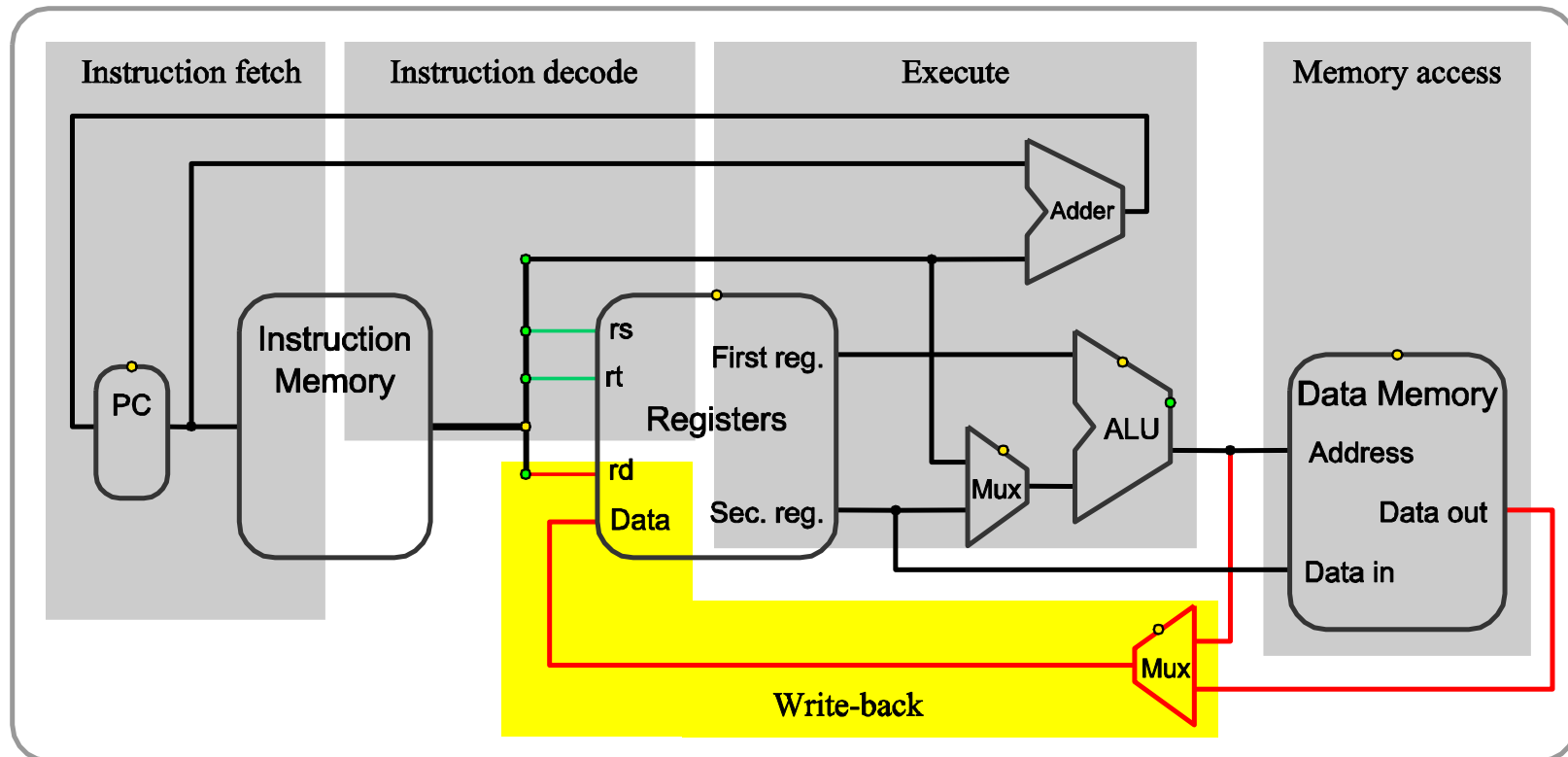
# De vijf fases van een instructie



# De vijf fases van een instructie



# De vijf fases van een instructie





# Toepassing: Pipelining

Methode om meer instructies tegelijkertijd uit te voeren

Voorbeeld: Baanvak Amsterdam Utrecht is onderverdeeld in 4 baanvakken

Tijd (min) Trein nr	6	12	18	24	30	36	42	48	54
1	CS-Amstel	Amstel-Abcoude	Abcoude-Breukelen	Breukelen-Utrecht					
2		CS-Amstel	Amstel-Abcoude	Abcoude-Breukelen	Breukelen-Utrecht				
3			CS-Amstel	Amstel-Abcoude	Abcoude-Breukelen	Breukelen-Utrecht			
4				CS-Amstel	Amstel-Abcoude	Abcoude-Breukelen	Breukelen-Utrecht		
5					CS-Amstel	Amstel-Abcoude	Abcoude-Breukelen	Breukelen-Utrecht	
6						CS-Amstel	Amstel-Abcoude	Abcoude-Breukelen	Breukelen-Utrecht
Treinen op diverse baanvakken									

4 baanvakken:  $(10 + 3) * 6 = 78$  min.

1 baanvak:  $10 * 4 * 6 = 240$  min.



# Toepassing: Pipelining

Tijd (ns) Instr. nr	0,2	0,4	0,6	0,8	1,0	1,2	1,4	1,6	1,8	2
1	Fetch	Dec.	Exe	Mem	WB					
2		Fetch	Dec.	Exe	Mem	WB				
3			Fetch	Dec.	Exe	Mem	WB			
4				Fetch	Dec.	Exe	Mem	WB		
5					Fetch	Dec.	Exe	Mem	WB	
6						Fetch	Dec.	Exe	Mem	WB

Executie van instructies in een pipeline machine

# Toepassing: Pipelining

Tijd (ns) Instr. nr	0,2	0,4	0,6	0,8	1,0	1,2	1,4	1,6	1,8	2
1	Fetch	Dec.	Exe	Mem	WB					
2		Fetch	Dec.	Exe	Mem	WB				
3			Fetch	Dec.	Exe	Mem	WB			
4				Fetch	Dec.	Exe	Mem	WB		
5					Fetch	Dec.	Exe	Mem	WB	
6						Fetch	Dec.	Exe	Mem	WB

Executie van instructies in een pipeline machine



# Toepassing: Pipelining

Tijd (ns) Instr. nr	0,2	0,4	0,6	0,8	1,0	1,2	1,4	1,6	1,8	2
1	Fetch	Dec.	Exe	Mem	WB					
2		Fetch	Dec.	Exe	Mem	WB				
3			Fetch	Dec.	Exe	Mem	WB			
4				Fetch	Dec.	Exe	Mem	WB		
5					Fetch	Dec.	Exe	Mem	WB	
6						Fetch	Dec.	Exe	Mem	WB

Executie van instructies in een pipeline machine

# Toepassing: Pipelining

Tijd (ns) Instr. nr	0,2	0,4	0,6	0,8	1,0	1,2	1,4	1,6	1,8	2
1	Fetch	Dec.	Exe	Mem	WB					
2		Fetch	Dec.	Exe	Mem	WB				
3			Fetch	Dec.	Exe	Mem	WB			
4				Fetch	Dec.	Exe	Mem	WB		
5					Fetch	Dec.	Exe	Mem	WB	
6						Fetch	Dec.	Exe	Mem	WB

Executie van instructies in een pipeline machine

# Toepassing: Pipelining

Tijd (ns) Instr. nr	0,2	0,4	0,6	0,8	1,0	1,2	1,4	1,6	1,8	2
1	Fetch	Dec.	Exe	Mem	WB					
2		Fetch	Dec.	Exe	Mem	WB				
3			Fetch	Dec.	Exe	Mem	WB			
4				Fetch	Dec.	Exe	Mem	WB		
5					Fetch	Dec.	Exe	Mem	WB	
6						Fetch	Dec.	Exe	Mem	WB

Executie van instructies in een pipeline machine



# Toepassing: Pipelining

Tijd (ns) Instr. nr	0,2	0,4	0,6	0,8	1,0	1,2	1,4	1,6	1,8	2
1	Fetch	Dec.	Exe	Mem	WB					
2		Fetch	Dec.	Exe	Mem	WB				
3			Fetch	Dec.	Exe	Mem	WB			
4				Fetch	Dec.	Exe	Mem	WB		
5					Fetch	Dec.	Exe	Mem	WB	
6						Fetch	Dec.	Exe	Mem	WB

Executie van instructies in een pipeline machine