


Computer Systems

Steven Moerman



1

---

---

---

---

---


---

---

---

H7 De hypothetisch x86

- De registers
  - Geheugenplaatsen in de processor
  - Geen onderdeel van hoofdgeheugen
  - X86 heeft 4 registers van 16-bits:
    - AX accumulator register
    - BX base address register
    - CX count register
    - DX data register
- Interne registers
  - Instruction Pointer (volgende instructie)
  - Vlaggenreger (resultaat vergelijking)



2

---

---

---

---

---


---

---

---

H7 De hypothetisch x86

- De ALU (arithmetic & logic unit)
  - Rekenkundige bewerkingen
  - Logische bewerkingen
- Bv 6 optellen bij AX-register
  - Kopieer de waarde van AX naar ALU
  - Stuur de waarde 6 naar de ALU
  - Geef opdracht aan ALU om een optelling uit te voeren
  - Breng resultaat terug naar het AX-register



3

---

---

---

---

---

---

---

---

### H7 De hypothetisch x86

- De bus interface unit (BIU) is verantwoordelijk voor:
  - Adres-bus
  - Data-bus
  - Cachegeheugen



4

---

---

---

---

---

---

---

### H7 De hypothetisch x86

- De control unit en de instructieset
  - Vaste verzameling instructies = instructieset
    - Logische poorten
  - Instructie + operand
- Pre-von Neumanr
 

	move	add	subtract	multiply	divide	and	or	xor
instr 1	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
instr 2	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
instr 3	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
...								

  - Patchkabels
  - Hard-wiring



5

---

---

---

---

---

---

---

### H7 De hypothetisch x86 p7-5

- Stored Program Concept
  - Control Unit
    - Cyclus:
      - Fetch
      - Execute



6

---

---

---

---

---

---

---

### H7 De hypothetisch x86 p7-5

- De MOV-instructie (verplaatsen doel <- bron)
  - MOV reg, reg/mem/const
  - MOV mem, reg
  - reg (Ax, Bx, Cx, Dx)
  - Const (hexadecimale waarde)
  - Mem (adres geheugengebied)



7

---

---

---

---

---

---

---

### H7 De hypothetisch x86 p7-5

- De rekenkundige en logische bewerkingen
  - ADD reg, reg/mem/const
  - SUB reg, reg/mem/const
  - CMP reg, reg/mem/const
  - AND reg, reg/mem/const
  - OR reg, reg/mem/const
  - NOT reg, mem



8

---

---

---

---

---

---

---

### H7 De hypothetisch x86 p7-5

- De "Control Transfer"-instructies (breken sequentie)
  - JA dest jump if above
  - JAE dest jump if above or equal
  - JB dest jump if below
  - JBE dest jump if below or equal
  - JE dest jump if equal
  - JNE dest jump if not equal
  - JMP dest jump unconditional
  - IRET dest return from an interrupt



9

---

---

---

---

---

---

---

### H7 De hypothetisch x86

- De I/O-instructies
  - GET (input keyb => AX)
  - PUT (execute AX)
- Overige
  - HLT (stop programma)
  - BRK (kan programma nadien terug starten)



10

---

---

---

---

---

---

---

### H7 De hypothetisch x86

- Vb. met registers:
  - MOV ax, ax      doet niets (kopieer ax naar ax)
  - MOV ax, bx      kopieer bx naar ax
  - MOV ax, cx      kopieer cx naar ax
  - MOV ax, dx      kopieer dx naar ax



11

---

---

---

---

---

---

---

### H7 De hypothetisch x86

- Vb. met constanten:
  - MOV ax, 25
  - MOV ax, 195
  - MOV ax, 2056
  - MOV ax, 1000



12

---

---

---

---

---

---

---

### H7 De hypothetisch x86

- Adresseringen
  - Direct  
MOV ax, [1000]
  - Indirect  
MOV ax, [bx]
  - Indexed  
MOV ax, [1000+bx]



13

---

---

---

---

---

---

---

---

### H7 De hypothetisch x86

- Processoren families
    - CISC (Intel)
    - RISC (andere)
  - Vb x86 (eenvoudiger voorstelling)
- Typische instructie (8-bits):  
i i i r r m m m



14

---

---

---

---

---

---

---

---

### H7 De hypothetisch x86

- Typische instructie (8-bits):  
i i i r r m m m

i	i	i	
0	0	0	special
0	0	1	or
0	1	0	and
0	1	1	cmp
1	0	0	sub
1	0	1	add
1	1	0	mov r,...
1	1	1	mov m,...

r	r	
0	0	AX
0	1	BX
1	0	CX
1	1	DX

m	m	m	
0	0	0	AX
0	0	1	BX
0	1	0	CX
0	1	1	DX
1	0	0	[BX]
1	0	1	[xxxx+BX]
1	1	0	[xxxx]
1	1	1	const



15

---

---

---

---

---

---

---

---

## H7 De hypothetisch x86

- Vb. MOV ax, bx = 1 byte = 1100 0001

i i i r r m m m  
1 1 0 0 0 0 0 1

i	i	i	
0	0	0	special
0	0	1	or
0	1	0	and
0	1	1	cmp
1	0	0	sub
1	0	1	add
1	1	0	mov r,...
1	1	1	mov m,...

r	r	
0	0	AX
0	1	BX
1	0	CX
1	1	DX

m	m	m	
0	0	0	AX
0	0	1	BX
0	1	0	CX
0	1	1	DX
1	0	0	[BX]
1	0	1	[xxxx+BX]
1	1	0	[xxxx]
1	1	1	const



16

---

---

---

---

---

---

---

---

## H7 De hypothetisch x86

- Vb. MOV ax, [1000]

i i i r r m m m + [1 0 0 0]  
1 1 0 0 0 1 1 0

i	i	i	
0	0	0	special
0	0	1	or
0	1	0	and
0	1	1	cmp
1	0	0	sub
1	0	1	add
1	1	0	mov r,...
1	1	1	mov m,...

r	r	
0	0	AX
0	1	BX
1	0	CX
1	1	DX

m	m	m	
0	0	0	AX
0	0	1	BX
0	1	0	CX
0	1	1	DX
1	0	0	[BX]
1	0	1	[xxxx+BX]
1	1	0	[xxxx]
1	1	1	const



17

---

---

---

---

---

---

---

---

## H7 De hypothetisch x86

- Vb. MOV ax, [1000]

i i i i r r m m m  
1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0  
C 6 0 0 1 0

i	i	i	
0	0	0	special
0	0	1	or
0	1	0	and
0	1	1	cmp
1	0	0	sub
1	0	1	add
1	1	0	mov r,...
1	1	1	mov m,...

r	r	
0	0	AX
0	1	BX
1	0	CX
1	1	DX

m	m	m	
0	0	0	AX
0	0	1	BX
0	1	0	CX
0	1	1	DX
1	0	0	[BX]
1	0	1	[xxxx+BX]
1	1	0	[xxxx]
1	1	1	const



18

---

---

---

---

---

---

---

---

## H7 De hypothetisch x86

- Vb. MOV ax, [1000]

i	i	i	r	r	m	m	16 bits																		
1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C				6			0				0				1				0						

16 bit adres na instructie.

De laagste orde byte komt onmiddellijk na de opcode van de instructie, gevolgd door de hoogste orde byte.



19

---

---

---

---

---

---

---

---

## H7 De hypothetisch x86

- Vb. MOV ax, [2000]

i	i	i	r	r	m	m	16 bits															
1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
C				6			0				0				2				0			

16 bit adres na instructie.

De laagste orde byte komt onmiddellijk na de opcode van de instructie, gevolgd door de hoogste orde byte.



20

---

---

---

---

---

---

---

---

## H7 De hypothetisch x86

- Oefening: MOV ax, bx

i	i	i	r	r	m	m	m
1	1	0	0	0	0	0	1
C		1					

i	i	i	
0	0	0	special
0	0	1	or
0	1	0	and
0	1	1	cmp
1	0	0	sub
1	0	1	add
1	1	0	mov r,...
1	1	1	mov m,...

r	r	
0	0	AX
0	1	BX
1	0	CX
1	1	DX

m	m	m	
0	0	0	AX
0	0	1	BX
0	1	0	CX
0	1	1	DX
1	0	0	[BX]
1	0	1	[xxxx+BX]
1	1	0	[xxxx]
1	1	1	const



21

---

---

---

---

---

---

---

---

## H7 De hypothetisch x86

- Oefening: MOV ax, [bx]

i	i	i	r	r	m	m	m
1	1	0	0	0	1	0	0
C				4			

i	i	i
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

r	r
0	0
0	1
1	0
1	1

m	m	m
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



22

---

---

---

---

---

---

---

---

## H7 De hypothetisch x86

- Oefening: MOV ax, 1256

i	i	i	r	r	m	m	m	16-bits															
1	1	0	0	0	1	1	1	0	1	0	1	0	1	1	0	0	0	0	1	0	0	1	0
C				7				5				6				1				2			

i	i	i
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

r	r
0	0
0	1
1	0
1	1

m	m	m
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



23

---

---

---

---

---

---

---

---

## H7 De hypothetisch x86

- Oefening: MOV [1234], ax

i	i	i	r	r	m	m	m	16-bits															
1	1	1	0	0	1	1	0	0	0	1	1	0	1	0	0	0	0	1	0	0	1	0	
E			6					3				4				1				2			

i	i	i
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

r	r
0	0
0	1
1	0
1	1

m	m	m
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



24

---

---

---

---

---

---

---

---



## H7 De hypothetisch x86

- Oefening: MOV [2579], ax

16-bits															
1	1	1	0	0	1	1	0	0	1	1	1	1	0	0	1
E				6				7				9			

i	i	i	r	r	m	m	m
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

r	r	m	m	m
0	0	0	0	0
0	1	0	0	0
1	0	0	0	0
1	1	0	0	0

m	m	m	m	m
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

25

## H7 De hypothetisch x86

- Special opcodes

16-bits															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

i	i	i	r	r	m	m	m
0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0

m	m	m	m	m
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

26

## H7 De hypothetisch x86

- Jump opcodes

16-bits															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

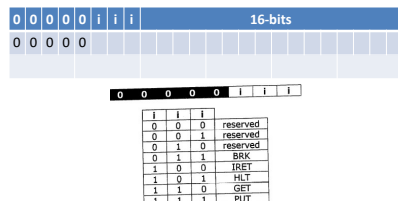
  

i	i	i	r	r	m	m	m
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

27

## H7 De hypothetisch x86

- Other opcodes



28

---

---

---

---

---

---

## H7 De hypothetisch x86

- **Stapsgewijs**
  - Fetch de instructie uit het geheugen (1 klokcyclus)
  - Pas de instructiepointer aan (1 klokcyclus)
  - Decodeer de instructie (1 klokcyclus)
  - Fetch indien nodig de 16 bits operand voor de instructie (0,1,2)
  - Pas indien nodig de instructiepointer aan (0,1)
  - Bereken indien nodig het adres van de operand (xxxx+bx) (0,1,2)
  - Fetch de operand (0,1,2,3)
  - Sla de waarde op in het doelregister (1 klokcyclus)

29

---

---

---

---

---

---

## H7 De hypothetisch x86

- De instructie MOV Mem, Reg
  - Fetch de instructie uit het geheugen
  - Pas de instructiepointer aan
  - Decodeer de instructie
  - Fetch indien nodig de operand voor de instructie
  - Pas indien nodig de instructiepointer aan
  - Bereken indien nodig het adres van de operand
  - Fetch de operand
  - Sla de waarde op in het doelregister

30

---

---

---

---

---

---

### H7 De hypothetisch x86

- De instructies ADD, SUB, CMP, AND, OR
  - Fetch de instructie uit het geheugen
  - Pas de instructiepointer aan
  - Decodeer de instructie
  - Fetch indien nodig de operand voor de instructie
  - Pas indien nodig de instructiepointer aan
  - Bereken indien nodig het adres van de operand
  - Fetch de waarde van de operand en stuur naar ALU
  - Fetch de waarde van de eerste operand (register) en stuur naar ALU
  - Geef opdracht aan ALU (add, sub, cmp, and of or)
  - Sla de waarde op in het doelregister



31

---

---

---

---

---

---

---

---

### H7 De hypothetisch x86

- De instructie NOT
  - Fetch de instructie uit het geheugen
  - Pas de instructiepointer aan
  - Decodeer de instructie
  - Fetch indien nodig de operand voor de instructie
  - Pas indien nodig de instructiepointer aan
  - Bereken indien nodig het adres van de operand
  - Fetch de waarde van de operand en stuur naar ALU
  - Geef de NOT opdracht aan ALU
  - Sla de waarde op in de operand



32

---

---

---

---

---

---

---

---

### H7 De hypothetisch x86

- De instructie Jxx
  - Fetch de instructie uit het geheugen
  - Pas de instructiepointer aan
  - Decodeer de instructie
  - Fetch indien nodig het doeladres uit het geheugen
  - Pas de instructiepointer aan
  - Test de kleiner dan of gelijk aan vlaggen
  - Als de vlaggen aan de gegeven conditie voldoen, kopieert de CPU de constante naar het IP-register



33

---

---

---

---

---

---

---

---

## H7 De hypothetisch x86

- Oefening p7-15

Machinetaal	Binaire Opcode	Operand	Addr
mov cx, 0	110 10 111	00 00	00, 01, 02
a: get	000 00 110		03
put	000 00 111		04
add ax, ax	101 00 000		05
put	000 00 111		06
add ax, ax	101 00 000		07
put	000 00 111		08
add ax, ax	101 00 000		09
put	000 00 111		0A
add cx, 1	101 10 111	01 00	0B, 0C, 0D
cmp cx, 4	011 10 111	04 00	0E, 0F, 10
jb a (= 0x0003)	000 01 010	03 00 = 8 bits mov, 8b per operand	
hlt	000 00 101		14

34

## H7 De hypothetisch x86

- Programma 1 p7-16

Machinetaal	Binaire Opcode	Operand	HEX	Addr
mov bx, 1000	110 01 111	00 10	CF-00-10	00, 01, 02
a: get	000 00 110		06	03
mov [bx], ax	111 00 100		E4	04
add bx, 2	101 01 111	02 00	AF-02-00	05, 06, 07
cmp ax, 0	011 00 111	00 00	67-00-00	08, 09, 0A
jne a	000 01 001	03 00	09-03-00	0B, 0C, 0D
mov cx, bx	110 10 001		D1	0E
mov bx, 1000	110 01 111	00 10	CF-00-10	0F, 10, 11
mov ax, 0	110 00 111	00 00	C7-00-00	12, 13, 14
b: add ax, [bx]	101 00 100		A4	15
add bx, 2	101 01 111	02 00	AF-02-00	16, 17, 18
cmp bx, cx	011 01 010		6A	19
jb b	000 01 010	15 00	0A-15-00	1A, 1B, 1C
put	000 00 111		07	1D
hlt	000 00 101		05	1E

35

## H7 De hypothetisch x86

- Programma 2 p7-16

Machinetaal	Binaire Opcode	Operand	HEX	Addr
a: get	000 00 110		06	00
cmp ax, 0	011 00 111	00 00	670000	01,02,03
jne b	000 01 001	0C 00	090C00	04,05,06
mov ax, [1000]	110 00 110	00 10	C60010	07,08,09
put	000 00 111		07	0A
halt	000 00 101		05	0B
b: cmp ax, [1000]	011 00 110	00 10	660010	0C,0D,0E
jbe a	000 01 011	00 00	0B0000	0F,10,11
mov [1000], ax	111 00 110	00 10	E60010	12,13,14
jmp a	000 01 110	00 00	0E0000	15,16,17

36