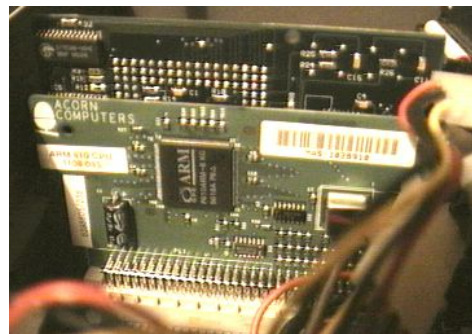


ARM-assembler cursus

Dick Tanis

2006

```
1580 : MOVNE R0,#7
1590 : LDRNE R2,[R12,#4]
1600 : SWINE "XOS_Module"
1610 : LDMFD R13!,{pc}
1620 :
1630 : .SUIHandler
1640 : MOV R10,R12
1650 : LDR R12,[R12,#0]
1660 : CMP R11,#10
1670 : ADDLT pc,pc,R11,LSL #2
1680 : B swi_unknown
1690 : B swi_resetdict
```



Dick Tanis

ARM-assembler cursus

6 mei 2006

Inhoud

- 1 6 mei 2006
 - Inleiding
 - Computer model
 - Interne architectuur ARM

Dick Tanis

ARM-assembler cursus

Inleiding cursus

Doel cursus:

- Werking van de ARM en architectuur
- Instructies + condities
- Data-verwerking en stacks
- ARM-assembler in basic
- Maken van utilities
- Maken van modules ?

Getalstelsels

Getallen zijn een som van machten: $n_0k^0 + n_1k^1 + n_2k^2 + \dots$,
Integers k en $n = 0, \dots, k - 1$

Belangrijkste getalstelsels:

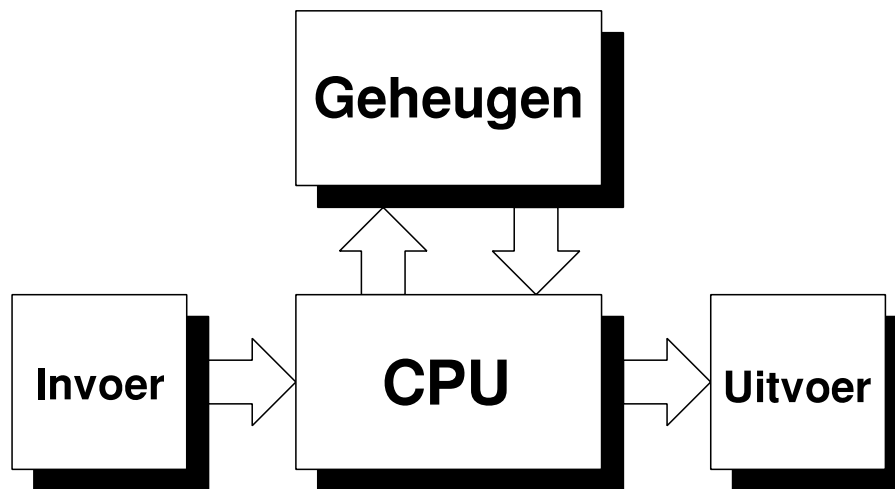
- Decimaal, $k = 10$: 3, 48, 345, 1454, 32678
- Binair, $k = 2$: %1, %10, %101, %1011, %10100
- Hexadecimaal, $k = 16$: &A, &1D, &E10, &1F4A, &3EFB

Reden hexadecimaal:

&F = %1111, &FF = %1111 1111

&A = %1010, &AF = %1010 1111

Typisch computer systeem



- Invoer/uitvoer
- Geheugen (in bytes onder RISC OS)
- Communicatie-bussen

Communicatie-bussen

Verbinding tussen CPU en andere andere delen in de computer

Data-bus: overbrengen data van en naar geheugen

- parallel
- 32 bits breed op de ARM, oftewel 1 word = 4 bytes
- bidirectioneel

Adres-bus: specificiëren van geheugen-locatie

- ARM 2 en 3, alleen 26 bits
- ARM 6 en hoger, 26 of 32 bits
- XScale alleen 32 bit

Word/byte locaties in geheugen

Benaderen van data in geheugen als 8 bits (byte) of 32 bits (word)

Bit 31.....0					
Locatie 0	byte 3	byte 2	byte 1	byte 0	word 0
Locatie 4	byte 7	byte 6	byte 5	byte 4	word 1
Locatie 8	byte 11	byte 10	byte 9	byte 8	word 2
Locatie 12	byte 15	byte 14	byte 13	byte 12	word 3

Ophalen van words in het geheugen kan alleen met **word-aligned** (deelbaar door 4) adressen

Instructies/processor cyclus

Instructies zijn binaire nummers en bestaan uit bitgroepen.

- Groep voor operaties: optellen, aftrekken etc
- Groep voor adressen in geheugen
- Groep voor registers
- Groep voor speciale operaties: shiften etc

Cyclus RISC processor in fases:

- 1 Ophalen instructie (Fetch)
- 2 Decoderen instructie en ophalen instructies (Decode)
- 3 Uitvoeren van instructie (Execute)
- 4 Toegang tot cache (Access)
- 5 Wegschrijven naar register/geheugen (Writeback)

Pipelining

In ARM 2-7, 3 fases onafhankelijk en overlapbaar

Cyclus 1	ophalen instructie 1	<leeg>	<leeg>
Cyclus 2	ophalen instructie 2	decoderen instructie 1	<leeg>
Cyclus 3	ophalen instructie 3	decoderen instructie 2	uitvoeren instructie 1
Cyclus 4	ophalen instructie 4	decoderen instructie 3	uitvoeren instructie 2

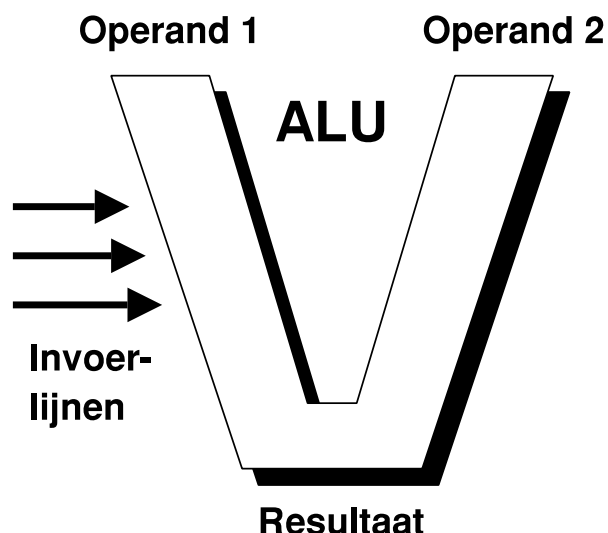
Voordeel: Effectief 1 instructie uitgevoerd per cyclus

Nadeel: Pipeline moet soms geleegd worden (branch-instructies)

Arithmetic Logic Unit (ALU)

Instructies:

- data-instructies: verplaatsen van registers, ARM ↔ geheugen
- rekenkundige instructies, ALU: optellen, aftrekken, vermenigvuldigen, OR, AND etc



Barrel shifter

Eerst gaat operand 2 via de barrel shifter naar de ALU. De barrel shifter verschuift data in operands.

Shift van 3 naar links

Data voor verschuiving: %0110100110011001111001011010111

Data na verschuiving : %0100110011001111001011010111000

Barrel shifter heeft 3 elementen nodig:

- 1 Originele operand die verschoven moet worden
- 2 Het aantal verschuivingen 0-32
- 3 Type verschuiving: links, rechts etc

Processor registers

Register:

Hardware element in de CPU

32 bits groot

Register ↔ geheugen

Toegang tot register is sneller dan tot geheugen

R0 tot en met R15

Speciale registers:

- R15: program counter (26/32 bit)/status register (26 bits)
- R14: link register (voor subroutines)
- CPSR: huidige status register (32 bit), ARM 6 en hoger
- SPSR: status register vorige modus (32 bit), ARM 6 en hoger

26 bit programm counter en status register

Register 15:

- status register, bit 31-28
- interrupt-vlaggen, bit 27 en 25
- geheugenadres voor de volgende instructie, bit 25-2
- modus-vlaggen, bit 1 en 0

Bit:	31	30	29	28	27	26	25		2	1	0
R15:	N	Z	C	V	I	F	...	Programm counter...	S1	S0	

Status vlaggen

De status vlaggen betekenen:

N	Negative	Is 1 bij een negatief resultaat van een berekening
Z	Zero	Is 1 als uitkomst van een berekening nul is
C	Carry	Bit 32 voor 64 bit berekeningen
V	Overflow	Is 1 als resultaat van een berekening niet in 32 bits past of bij een foutmelding (RISC OS)
I	IRQ	Geen interrupts dan 1
F	FIQ	Geen snelle interrupts dan 1

Modus vlaggen

S1 en S0 leveren de volgende processor modi op:

S1	S0	Modus		Functie
0	0	USR	Gebruiker modus	Hierin draaien de applica- ties
0	1	FIQ	Snelle inter- rupt modus	Afhandelen van externe devices
1	0	IRQ	Interrupt modus	Afhandelen van externe devices
1	1	SVC	Supervisor modus	OS-routines draaien in deze modus, SWI's

Registers in diverse modi

Register Naam	Processor registers in modus:			
	User	FIRQ	IRQ	SVC
R0	R0	R0	R0	R0
R1	R1	R1	R1	R1
R2	R2	R2	R2	R2
R3	R3	R3	R3	R3
R4	R4	R4	R4	R4
R5	R5	R5	R5	R5
R6	R6	R6	R6	R6
R7	R7	R7	R7	R7
R8	R8	R8_FIRQ	R8	R8
R9	R9	R9_FIRQ	R9	R9
R10	R10	R10_FIRQ	R10	R10
R11	R11	R11_FIRQ	R11	R11
R12	R12	R12_FIRQ	R12	R12
R13	R13	R13_FIRQ	R13_IRQ	R13_SVC
R14	R14	R14_FIRQ	R14_IRQ	R14_SVC
R15	R15	R15	R15	R15