

Let op:

Vul het tentamenbriefje volledig in (d.w.z. naam, studentnummer, naam vak, vakcode, naam docent “Kokkeler/Molenkamp/Scholten”) en zorg er ook voor dat de kopieën goed leesbaar zijn.

Aanwijzingen bij het maken van het tentamen:

1. Beantwoord de vragen uitsluitend op de aangegeven plaatsen op dit vragenformulier
2. Vul op de eerste bladzijde uw naam, opleiding en studentnummer in.
3. Vul op de overige bladzijden uw naam in.
4. Lever altijd alle bladzijden in (documentatie mag worden meegenomen).
5. Tijdens dit tentamen is alleen het gebruik van schrijfgerei en een rekenmachine toegestaan. Andere schriftelijke bronnen en apparatuur (organizers, laptops, etc.) zijn verboden.

Naam:

Studentnummer:

Opleiding:

OPGAVE 1

(5 + 5 = 10 PUNTEN)

a) In de documentatie van de ARC processor is op pagina 3 (fig. 5-10) het datapad en besturing gegeven. Hierin komt o.a. een ALU voor. Afhankelijk van de ALU operatie (pagina 2, fig. 5-4) moeten de conditie code bits (%psr) worden aangepast. Als %psr aangepast moet worden is uitgang *Set Condition Codes* (scc) ‘1’. Als de %psr niet wordt beïnvloed door de operatie is scc gelijk aan ‘0’. Gevraagd wordt een minimale booleaanse vergelijking in SOP (sum-of-products) voor scc.

$$scc = !F3.!F2$$

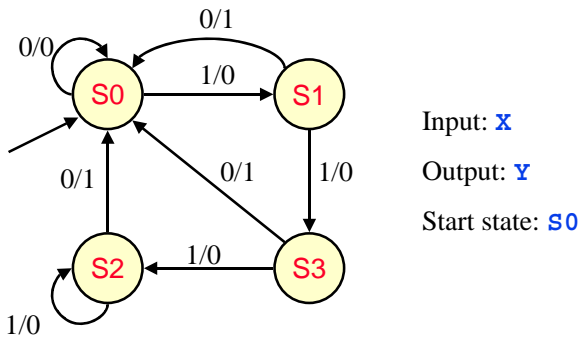
b)

	CD				
		00	01	11	10
AB	00	1	0	0	1
	01	0	0	0	0
	11	0	1	1	0
	10	1	1	1	1

Geef voor bovenstaand Karnaugh diagram (K-map) voor functie F een minimale logische formule in sum-of-products.

$$F = !B.!D + A.D$$

a) Gegeven het volgende toestandsdiagram.



Vul voor het gegeven toestandsdiagram deze tabel in.

Present State	Next state		Output	
	X=0	X=1	X=0	X=1
			Y	Y
S0	S0	S1	0	0
S1	S0	S3	1	0
S2	S0	S2	1	0
S3	S0	S2	1	0

b) Hoeveel flipflops zijn er minimaal nodig om dit circuit te realiseren?

2
1 (omdat S1 en S3 niet nodig zijn; minimalisatie echter geen onderdeel tentamen)

c) Het systeem wordt gerealiseerd met vier D-flipflops: F3, F2, F1 en F0. Er wordt gebruik gemaakt van de one-hot codering; d.w.z. exact 1 van de flipflop uitgangen is '1' en de andere zijn '0'. De codering is als volgt (resp. F3, F2, F1, F0):

S0=0001, S1=0010, S2=0100, S3=1000

Gevraagd een minimale booleaanse expressie voor uitgang Y.

$Y = !F0 \cdot !X$

Geef voor het decimale getal -12 het **binaire patroon met 8 bits** voor de aangegeven representaties:

Sign magnitude	10001100
1-complement	11110011
2-complement	11110100

Naam:

OPGAVE 4

(4 × 4 = 16 PUNTEN)

Gegeven is een genormaliseerde floating point representatie:

- Sign bit: 1 bit (meest linker positie), gevolgd door
- Exponent veld: 4 bits in excess 8 code, grondtal exponent is 2, en ten slotte
- Mantissa veld: 5 bits (exclusief het hidden bit). De punt is links van het hidden bit.

Alleen als het exponent veld gevuld is met allemaal nullen worden gedenormaliseerde getallen voorgesteld (d.w.z. hidden bit is 0 en de decimale waarde van de exponent is gelijk aan de decimale waarde van de exponent van het kleinste genormaliseerde getal).

Geef bij de antwoorden een geheel getal, of een afgerond decimaal floating point getal met 1 cijfer voor en 5 cijfers achter de komma.

De vragen a) t/m c) beantwoorden voor de genormaliseerde representaties.

- a) Wat is de maximale decimale waarde van de mantissa (M_{max})

$$1-2^{-6} \sim 0.984375$$

- b) Wat is de minimale decimale waarde van de exponent (E_{min})

$$-7$$

- c) Wat is het grootste positieve decimale getal dat gerepresenteerd kan worden?

$$(1-2^{-6}) \times 2^7 = 126$$

- d) Wat is de decimale waarde van de het floating point getal 1 1011 11010

$$-(29/32) \times 8 = -7,25$$

Deze opgave heeft betrekking op de subset ARC processor (dus alleen instructies uit figuur 5-2 van de documentatie mogen worden gebruikt). De processor wordt gebruikt in een toepassing waarbij gebruik moet worden gemaakt van een zogenaamde ‘saturated add’, d.w.z. als het resultaat van de optelling groter is dan het grootste getal dat kan worden weergegeven, dan wordt het resultaat van de optelling het grootste positieve getal, evenzo als het resultaat kleiner is dan het meest negatieve getal dat kan worden weergegeven dan is het meest negatieve getal het resultaat van de optelling. Er wordt gebruik gemaakt van de 2-complementrepresentatie.

- a) Geef het decimale bereik van een 32 bits 2-complement getal? (machten van 2 in het antwoord zijn toegestaan)

-2^{31} t/m $2^{31}-1$

- b) Geef aan hoe je met maximaal twee assembly instructies het grootste positieve getal in register %r1 kan zetten (instructie *ld* mag niet worden gebruikt).

```
sethi 0x1FFFFFF, %r1
or   %r1, 0x3FF, %r1
```

alternatieven zijn mogelijk zolang maar geen data van het geheugen wordt gelezen.

- c) In deze vraag wordt aangenomen dat in %r1 het grootste positieve getal zit en in %r2 het meest negatieve getal. Schrijf een efficiënt stukje assembly dat uitgevoerd moet worden na de instructie *addcc %r3, %r4, %r5* om de *saturated add* functionaliteit te krijgen. Na deze toevoeging gaat het programma door met de regel met label *cont*. Van de registers %r0 t/m %r31 mag alleen de inhoud van %r5 worden gewijzigd. Ook de inhoud van het main memory moet ongewijzigd blijven. Voorzie je assembly van duidelijk commentaar!

```
.begin
org 0
sethi 0x1FFFFFF, %r1
or   %r1, 0x3FF, %r1 ! in %r1 staat nu max. 2 compl.
sethi 0x200000, %r2 ! in %r2 staat nu min. 2 compl.
addcc %r3, %r4, %r5
bvs  overf
ba  cont
overf: bneg max ! indien neg, dan max 2 compl
addcc %r2, %r0, %r5 ! min in %r5
ba  cont
max:  addcc %r1, %r0, %r5 ! max in %r5 ! of andcc %r1, %r1, %r5
```

ongetwijfeld ook alternatieven mogelijk.

```
cont: halt
.end
```

De ARC processor wordt uitgebreid met de ABS instructie.

ABS %rx, %ry met %ry ← absolute waarde van %rx, representatie 2 complement
 De *condition codes* mogen veranderd worden.
 %rx en %ry zijn registers van de register file (%r0 t/m %r31),

De opbouw van deze instructie (zie documentatie pagina 1, onderaan):
 op=10, rd=%ry, op3=101010, rs1=wordt niet gebruikt, bit13=0,
 bit12 t/m bit5=00000000, rs2=%rx

a) Het bepalen van het 2-complement van een operand wordt door meerdere instructies gebruikt. Daarom besluit het ontwerpteam om in het microgeheugen vanaf het decimale adres 500 een microprogramma te plaatsen dat het 2-complement bepaalt van de inhoud van register %temp0 en het resultaat komt in het register aangegeven door %IR[rd]. Van de registers %r0 t/m %r31 mag alleen dit resultaatregister worden gewijzigd. Ook de inhoud van het main memory moet ongewijzigd blijven. Na afloop wordt naar microadres 2047 gesprongen. Gevraagd wordt het microgeheugen in te vullen voor deze 2-complement bewerking. U mag symbolische 'namen' gebruiken (bijv. voor veld A ipv 00110 mag %r6, en veld ALU ipv 0000 mag ADDCC etc.). Velden die niet worden gebruikt vullen met '-' (don't care).

adres	A	Amux	B	Bmux	C	Cmux	Rd	Wr	ALU	Cond	Jump addr
500	%r0	0	%temp0	0	%temp0	0	0	0	orn	next 000	-
501	%temp0	0	-	-	-	1	0	0	INCA	jmp 110	2047

b) Tijdens het decoderen van de instructie ABS wordt het startadres bepaald in het microgeheugen. Wat is het startadres van deze instructie (decimale waarde geven).

11010101000₂ = 1704 (documentatie pagina 3)

c) Geef een efficiënt microprogramma voor de instructie ABS. Indien u het startadres (vraag b) niet weet mag u daarvoor 600 nemen. U mag aannemen dat op het decimale adres 500 een microprogramma staat met de werking zoals is beschreven onder vraag a). Ook nu weer don't cares ('-') gebruiken voor velden die niet worden gebruikt.

adres	A	Amux	B	Bmux	C	Cmux	Rd	Wr	ALU	Cond	Jump addr
1704	%r0	0	-	1	%temp0	0	0	0	Orcc 0001	nxt 000	-
1705	-	-	-	-	%r0	0	0	0	-	N=1 001	500
1706	%r0	0	-	1	-	1	0	0	orcc-	jmp 110	2047

Toelichting uitwerking:

Adres 1704: %temp ← %r0 orcc rs2; hier MOET orcc worden gebruikt voor de conditiecode bits
 Adres 1705: hier is pas het statusregister aangepast en kan op basis van N-bit worden gesprongen. Als N=1 dan wordt gesprongen naar uitwerking opgave a) merk op dat de operand al in %temp0 staat.
 Adres 1706: blijkbaar is operand als positief en wordt rd ← %r0 orcc %rs2 klaar.

OPGAVE 7:**(4 × 3 = 12 PUNTEN)**

Een embedded computersysteem heeft een fysisch adresbereik van 16 MB en een 32-bits databus. Het geheugen is byte-adresseerbaar. Het systeem bevat ROM en RAM volgens de volgende specificaties:

ROM: 1 MB op de laagste adressen van het adresbereik.

RAM: 8 MB op de hoogste adressen van het adresbereik.

Schaduwgebieden zijn niet toegestaan.

De selectiesignalen voor deze gebieden zijn respectievelijk SelROM en SelRAM.

- a) Geef in hexadecimale notatie het hoogste adres van het ROM-gebied.

0FFFFFFH

- b) Geef in hexadecimale notatie het **laagste** adres van het RAM-gebied.

800000H

- c) SelROM is een functie van een aantal adreslijnen.
Geef de logische formule voor SelROM.

SelROM = !A23&!A22&!A21&!A20

- d) SelRAM is een functie van een aantal adreslijnen.
Geef de logische formule voor SelRAM.

SelRAM = A23

Naam:

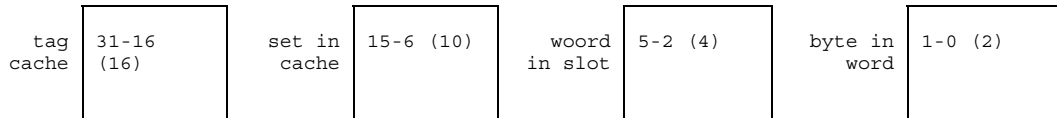
OPGAVE 8:

(3 × 3 + 2 × 3 = 15 PUNTEN)

Een 32-bits microprocessor heeft een cache met de volgende gegevens:

Fysisch adresbereik:	4 GB
Byte-adressering	
Cache: Grootte	256 kB data
Blok grootte	16 woorden (1 woord is 32 bits).
Organisatie	4-voudig set-associatief

- a) Voor de cache geldt dat een byte-adres is gesplitst in delen die worden gebruikt voor respectievelijk de vergelijking met de tag in de cache, de selectie van een set in de cache, de selectie van een woord in een slot en de selectie van een byte in een woord. Welke bitnummers horen bij elk van deze delen?



- b) Wat is het slotnummer binnen de cache, behorend bij de volgende adressen. De adressen worden gegeven in hexadecimale notatie:

- FFFFFFFF H
- 00012345 H

