

Tentamen Systeemarchitectuur 1

Eerste deel

22 mei 2008

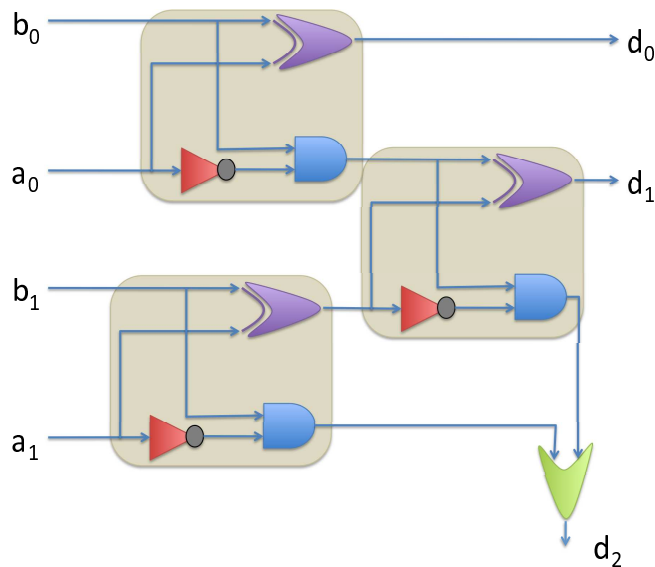
Beredeneer je antwoord. Vergeet de opgaven op de achterkant niet. Cijfer \approx punten / 8.

1. (8 P.) Er zijn verschillende mogelijkheden om negatieve getallen binair te coderen. „Offset-128” is een codering waarbij je 128 bij het getal optelt en dan het resultaat naar binair vertaalt. Voorbeelden: 13_{10} is in offset-128-codering $13_{10} + 128_{10} = 141_{10} = 10001011_2$ en -85_{10} wordt gecodeerd als $-85_{10} + 128_{10} = 43_{10} = 00101011_2$.

Geef aan hoe je op basis van één standaard-8-bit-adder een adder voor twee 8-bits-getallen in offset-128-notatie kunt bouwen. Beschrijf het principe en teken een circuit.

Extra opgave: (2 P.) Geef aan hoe de vlaggen daarna gezet moeten worden (carry, sign, zero, overflow). Je mag aannemen dat de standaard-adder de standaard-vlaggen genereert. Zijn alle vlaggen zinvol?

2. (6 P.) Welke functie heeft het circuit in figuur 1? (Let op: er is een klein verschil tussen de gate rechts onder en de gates in de drie kastjes.)



Figuur 1: Circuit dat bij opgave 2 hoort

3. (10 P.) Construeer een circuit dat de „pariteit” van een vier-bits-getal berekent, d.w.z. de uitvoer is 1 dan en slechts dan als de invoer een even aantal 1-bits heeft. Voorbeelden:

Invoer	Uitvoer
0000	1
0010	0
1010	1
1111	1

4. (5 P.) Een bepaalde processor heeft instructies van 16 bits. Sommige instructies hebben één operand, sommige twee. Om aan te geven waar een operand staat zijn 6 bits nodig. Als de processor n instructies met twee operanden heeft, hoeveel instructies met één operand kan hij maximaal hebben?
5. (5 P.) Beschouw het binaire getal $1001\ 0101\ 1100\ 0011_2$. Wat is het resultaat van de volgende berekeningen met dit getal:
 - (a) Logische shift right om 4 bits.
 - (b) Arithmetische shift right om 4 bits.
 - (c) Shift left om 4 bits.
 - (d) Rotate left om 4 bits.
 - (e) Rotate right om 4 bits.
6. (6 P.) Noem de drie belangrijkste soorten van assembly instructies en geef voor elke soort een voorbeeld in jouw favoriete assembly.
7. (12 P.) Schrijf een assembly-programma (in de assembly voor onze processor SA1/08 of in 8086-assembly) dat het volgende doet: bereken $a + b - 13$ en sla het resultaat op in a . Variabele a is op geheugenlocatie 10 opgeslagen en variabele b op locatie 12. Beide variabelen zijn 16 bit groot.
8. (8 P.) Bedenk een manier om de inhoud van twee registers R_a en R_b uit te wisselen zonder een ander register of het geheugen te gebruiken. *Tip:* Denk aan de XOR-instructie.
9. (10 P.) Aan het einde van een assembly-procedure moeten de volgende dingen gebeuren:
 - locale variabelen van de stack verwijderen
 - framepointer op zijn vroegere waarde zetten
 - terugspring-adres lezen
 - parameters van de stack verwijderen
 - ... en nog één taak

Welke taak ontbreekt in het lijstje? Welke van deze taken kan het oproepende programma doen? Wanneer is het zinvol dat niet de procedure, maar het oproepende programma sommige van deze taken doet?

10. (8 P.) Schrijf je naam en studentnummer op alle vellen die je inlevert.