

AoC1 undersøkelse høsten 2006 – Avsluttende test

Noter test-person-nr.her: [_____]

Du har 20 minutter på testen. Den er delt opp i tre ulike deler med ulike typer oppgaver. Dersom det skulle bli lite tid, forsøk å løse minst en oppgave i alle de tre delene.

Del A) Tema: Busser. Spørsmålstype: flervalg

Sett kryss på det ene alternativet du mener er rett svar. Riktig svar gir 1 poeng, flere kryss gir uansett 0 poeng.

A1) Hvilket av alternativene er IKKE en grunn til å ha **hierarkiske busser** i en datamaskin?

- Det er ønskelig å skille trege enheter fra raske enheter
- En lang buss får gjerne høy transmisjonsforsinkelse
- Med flere busser kan flere enheter overføre data samtidig
- Det kan være en fordel med spesialiserte busser mot enkelte enheter
- Lokalitetsprinsippet gjør det mindre sannsynlig at fjerne enheter blir brukt

A2) En multiplekset buss:

- Krever generelt færre ledere enn en dedikert buss
- Gir generelt høyere ytelse enn en dedikert buss
- Kan ikke brukes utenfor maskinen
- Gjør arbitrerer mer komplisert enn en dedikert buss
- Kan ikke brukes med moderne RAM-teknologi

A3) Hvilket av disse utsagnene er IKKE riktig?

- Arbitrerer er unødvendig på en synkron buss
- En buss er et delt transmisjonsmedium
- En buss er den vanligste sammenkoblingsstrukturen i en datamaskin
- En buss består typisk av kontroll, adresse og datalinjer
- Arbitrerer er ikke nødvendig dersom kun en av enhetene tilkoblet bussen kan være bussmaster

A4) Hva er sant når det gjelder kommunikasjon over en buss?

- Flere enheter kan kommunisere med hverandre samtidig hvis de er ved siden av hverandre på bussen
- En enhet kan bestandig sende til en annen enhet så lenge ikke den andre enheten sender et signal på bussen
- Busser går bare en vei
- Bare en enhet kan legge signal på bussen av gangen

A5) Hva er rett når det gjelder arbitrerer?

- Under arbitrerer kodes signalene som skal sendes over bussen
- Under arbitrerer bestemmes hvilken enhet som skal få bruke bussen
- Arbitrerer verifiserer at alle lederne i bussen fungerer normalt
- Arbitrerer skyldes forsinkelsen over bussen
- Arbitrerer brukes for å øke ytelsen på bussen

A6) En gitt buss med tre eller flere enheter tilkoblet inneholder 16 datalinjer og 16 adresselinjer. Disse kan brukes til å:

- Opprette 2^{16} forskjellige forbindelser som hver kan overføre 16 bit
- Overføre 16 bit i hver retning samtidig mellom to enheter
- Samtidig overføre 16 bit mellom to enheter
- 4 forbindelser mellom forskjellige enheter, der en overfører 1 byte samtidig på hver forbindelse
- Overføre et multiplum av 16 bit mellom to enheter. Hvor mange bit som bussen kan håndtere avhenger av lengden av bussen

Del B, Tema: Adressemodi. Spørsmålstype: Tallverdi

Angi riktig svar som en tallverdi i svarfeltet. Riktig svar gir 2 poeng.

Gitt en maskin med et antall registre R0, R1, R2 osv, (benevnt 'Rn' nedenfor), og et datalager. Startverdier for lager, registre og noen variable er vist i høyre del av figuren under.

Instruksjonen MOV er definert slik: MOV til-operand, fra-operand. De forskjellige adresseringsmodi noteres som vist i venstre del av figuren under:

MOV Rn, #Imm ; Immediate	<table border="1"> <thead> <tr> <th colspan="2">Hovedlager</th> <th colspan="2">Registre</th> <th colspan="2">Variable</th> </tr> <tr> <th>verdi</th> <th>adr</th> <th>reg</th> <th>verdi</th> <th>navn</th> <th>adr</th> </tr> </thead> <tbody> <tr> <td>233</td> <td>7</td> <td>R0</td> <td>0</td> <td>VAR1</td> <td>16</td> </tr> <tr> <td>15</td> <td>8</td> <td>R1</td> <td>1</td> <td>VAR2</td> <td>8</td> </tr> <tr> <td>4</td> <td>9</td> <td>R2</td> <td>4</td> <td>VAR3</td> <td>10</td> </tr> <tr> <td>15</td> <td>10</td> <td>R3</td> <td>3</td> <td>TABELL</td> <td>11</td> </tr> <tr> <td>17</td> <td>11</td> <td>R4</td> <td>15</td> <td></td> <td></td> </tr> <tr> <td>12</td> <td>12</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>17</td> <td>13</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>9</td> <td>14</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>7</td> <td>15</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>8</td> <td>16</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>21</td> <td>17</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>19</td> <td>18</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>7</td> <td>19</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>9</td> <td>20</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Hovedlager		Registre		Variable		verdi	adr	reg	verdi	navn	adr	233	7	R0	0	VAR1	16	15	8	R1	1	VAR2	8	4	9	R2	4	VAR3	10	15	10	R3	3	TABELL	11	17	11	R4	15			12	12					17	13					9	14					7	15					8	16					21	17					19	18					7	19					9	20					
Hovedlager		Registre		Variable																																																																																														
verdi		adr	reg	verdi	navn	adr																																																																																												
233		7	R0	0	VAR1	16																																																																																												
15		8	R1	1	VAR2	8																																																																																												
4		9	R2	4	VAR3	10																																																																																												
15		10	R3	3	TABELL	11																																																																																												
17		11	R4	15																																																																																														
12		12																																																																																																
17	13																																																																																																	
9	14																																																																																																	
7	15																																																																																																	
8	16																																																																																																	
21	17																																																																																																	
19	18																																																																																																	
7	19																																																																																																	
9	20																																																																																																	
MOV Rn, Addr ; Direkte																																																																																																		
MOV Rn, [Addr] ; Indirekte																																																																																																		
MOV Rn, Rm ; Register																																																																																																		
MOV Rn, [Rm] ; Register indirekte																																																																																																		
MOV Rn, Addr(Rm) ; Displacement																																																																																																		
PUSH Rn ; Stakk push																																																																																																		
POP Rn ; Stakk pop																																																																																																		

B1) Hva vil verdien til R1 være etter følgende instruksjon: MOV R1, 17? Svar her [_____]

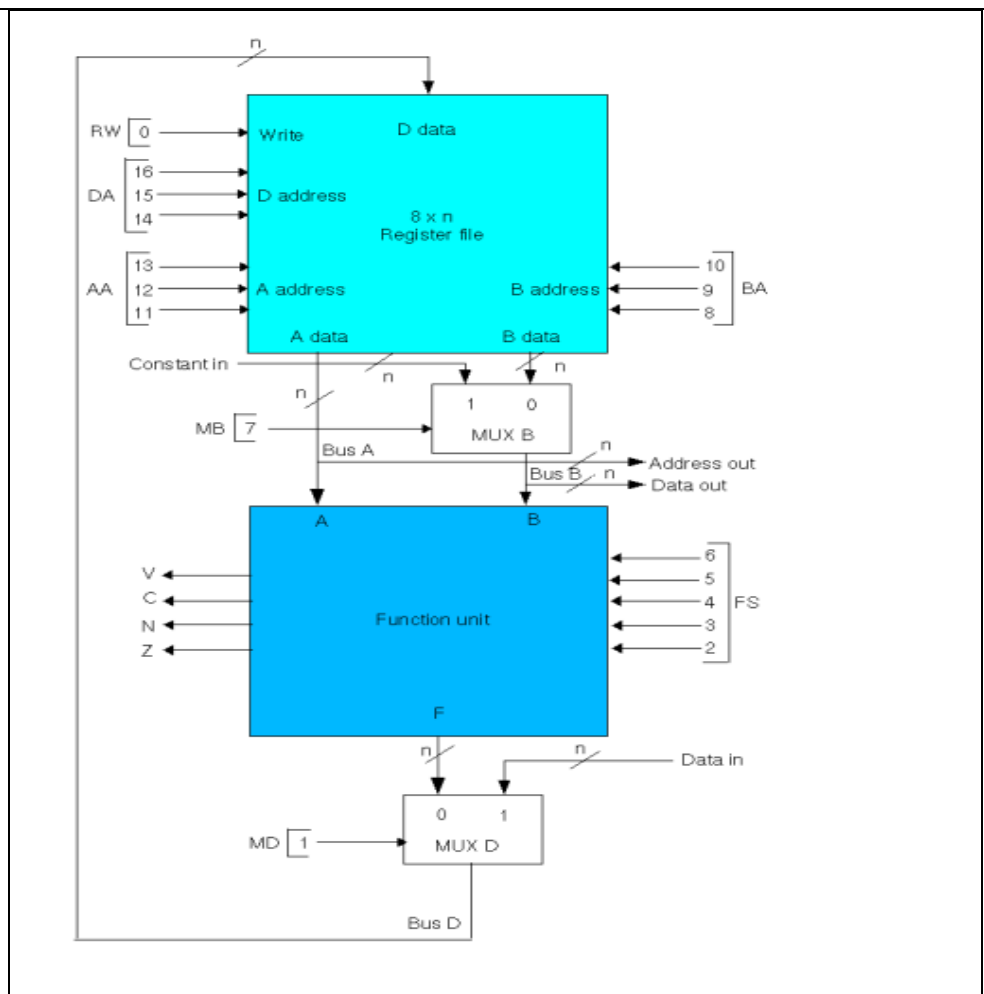
B2) Hva vil verdien til R1 være etter følgende instruksjon: MOV R1, #9? Svar her [_____]

B3) Hva vil verdien til R1 være etter følgende instruksjon: MOV R1, 8(R2)? Svar her [_____]

Del-C, Tema: Styreord. Spørsmålstype: Bit-vektor

Utførende enhet kontrolleres av et styreord. Styreordet er bare en samling av alle styresignalene til den utførende enheten. I disse oppgavene skal du sette opp styreordet slik at du får utførende enhet til å gjøre det du vil. Figur 1 viser utførende enhet. Du skal sette opp styreordet slik at den oppgitte mikrooperasjonen blir utført. Bruk styreordformatet vist i figur 2 og 3. Husk at register 0 inneholder tallet 0. Konstanter kommer fra instruksjonsordet og skal derfor ikke settes opp i styreordet. Du kan anta at dette er riktig satt opp. Svar på spørsmålet ved å bruke opplysningene over. Angi svaret binært og bruk x for verdier som ikke brukes (Don't care). $RD \leftarrow RA$ op RB betyr at register RD tilordnes verdien av funksjonen 'op' med RA og RB som operander. Funksjoner med kun en operand bruker bare RA .

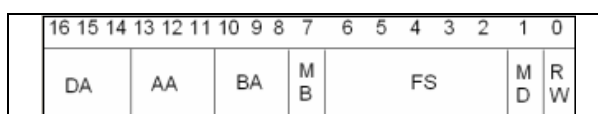
Opgavene står nede til høyre. Riktig svar gir 3 poeng pr. oppgave.



Figur 1. Styreenhet

DA, AA, BA		MB		FS		MD		RW	
Function	Code	Function	Code	Function	Code	Function	Code	Function	Code
$R0$	000	Register	0	$F = A$	00000	Function	0	No Write	0
$R1$	001	Constant	1	$F = A + 1$	00001	Data In	1	Write	1
$R2$	010			$F = A + B$	00010				
$R3$	011			$F = A + B + 1$	00011				
$R4$	100			$F = A + \sim B$	00100				
$R5$	101			$F = A + \sim B + 1$	00101				
$R6$	110			$F = A - 1$	00110				
$R7$	111			$F = \sim A$	00111				
				$F = A \wedge B$	01000				
				$F = A \vee B$	01010				
				$F = A \text{ XOR } B$	01100				
				$F = A$	01110				
				$F = sr A$	10000				
				$F = sl A$	10001				

Figur 2: Binære koder for bruk i styreord.



Figur 3: 17 bits styreordformat.

C1) Hva blir styreordet for operasjonen

$R7 \leftarrow R1 + \sim R5 + 1$
der x beskriver ubrukte bits?

Svar her:

[_____]

C2) Hva blir styreordet for operasjonen

$R7 \leftarrow R2 \text{ XOR } R0$
der x beskriver ubrukte bits?

Svar her:

[_____]

Takk for hjelpen!