Tentamen

Datorteknik Y, TSEA28

Datum	2017-08-15
Lokal	TER4
Tid	14-18
Kurskod	TSEA28
Provkod	TEN1
Kursnamn *Provnamn*	Datorteknik Y Skriftlig tentamen
Institution	ISY
Antal frågor	6
Antal sidor (inklusive denna sida)	6
Kursansvarig	Kent Palmkvist, 1347, kentp@isy.liu.se
Lärare som besöker skrivsalen *Telefon under skrivtiden*	Kent Palmkvist 1347, 013-281347
Besöker skrivsalen	Ca kl 15 och kl 17
Kursadministratör	Gunnel Hässler
Tillåtna hjälpmedel	Inga
Betygsgränser	Preliminärt 0-20: U, 21-30: 3, 31-40: 4, 41-50: 5
Visning	Onsdag 30 Augusti kl 12.30 – 14.00 i kursansvarigs kontor

Viktig information

- För de uppgifter där du måste göra uträkningar är det viktigt att du redovisar alla relevanta mellansteg
- I de uppgifter där du ska skriva assembler- eller mikrokod är det viktigt att du kommenterar koden
- Om du i en viss uppgift ska förklara något, tänk på att du gärna får rita figurer för att göra din förklaring tydligare
- Uppgifterna i tentan är inte ordnade efter svårighetsgrad
- Skriv inga svar i detta häfte!

Lycka till!

Fråga 1: Mikroprogrammering (10p)

I figur 1 återfinns en bekant datormodell som du ska mikroprogrammera i denna uppgift.

Jämfört med den modell ni sett tidigare har följande förändring gjorts: Mikroprogrammeringsordet har utökats med ytterligare en bit (kallad R). Om R=0 fungerar datorn precis som tidigare. Om R=1 sätts styrsignaler som styr multiplexern vid GR0-GR3 till 3 (dvs register GR3 väljs), oavsett vad IR innehåller.

Notera att det är viktigt att du skriver vilken additionsoperation du valt (den som påverkar flaggorna eller den som inte påverkar flaggorna). (Om du inte skriver något speciellt kommer jag att anta att du ej vill modifiera flaggorna).

Mikrokodsminnnet innehåller från början

addr	Mikrokod	Kommentar
0	ASR := PC, uPC := uPC+1	
1	IR := PM, PC := PC+1, uPC := uPC+1	
2	uPC := K2(M)	
3	ASR := IR, uPC := K1(OP)	; direktadressering (M=00)
4	ASR := PC, PC := PC+1, uPC := K1(OP)	; omedelbar adressering (M=01)
5	ASR := IR, uPC := uPC+1	; indirekt adressering (M=10)
6	ASR := PM, uPC := K1(OP)	
7	PM := GRx, S := 0, R := 0, uPC := 0	; STORE
8	uPC := 0 om Z = 1, annars uPC++	; JNE addr
9	PC := IR, uPC := 0	
10	GRx := PM, S := 0, R := 0, uPC := 0	; LOAD

(a) (6p) Skriv mikrokod för instruktionen DECJNZ GRx,A (decrement GRx, jump to A if non zero). Register GRx ska minskas med 1, och om resultatet inte är 0 ska ett hopp utföras till adressen i instruktionens A-fält.

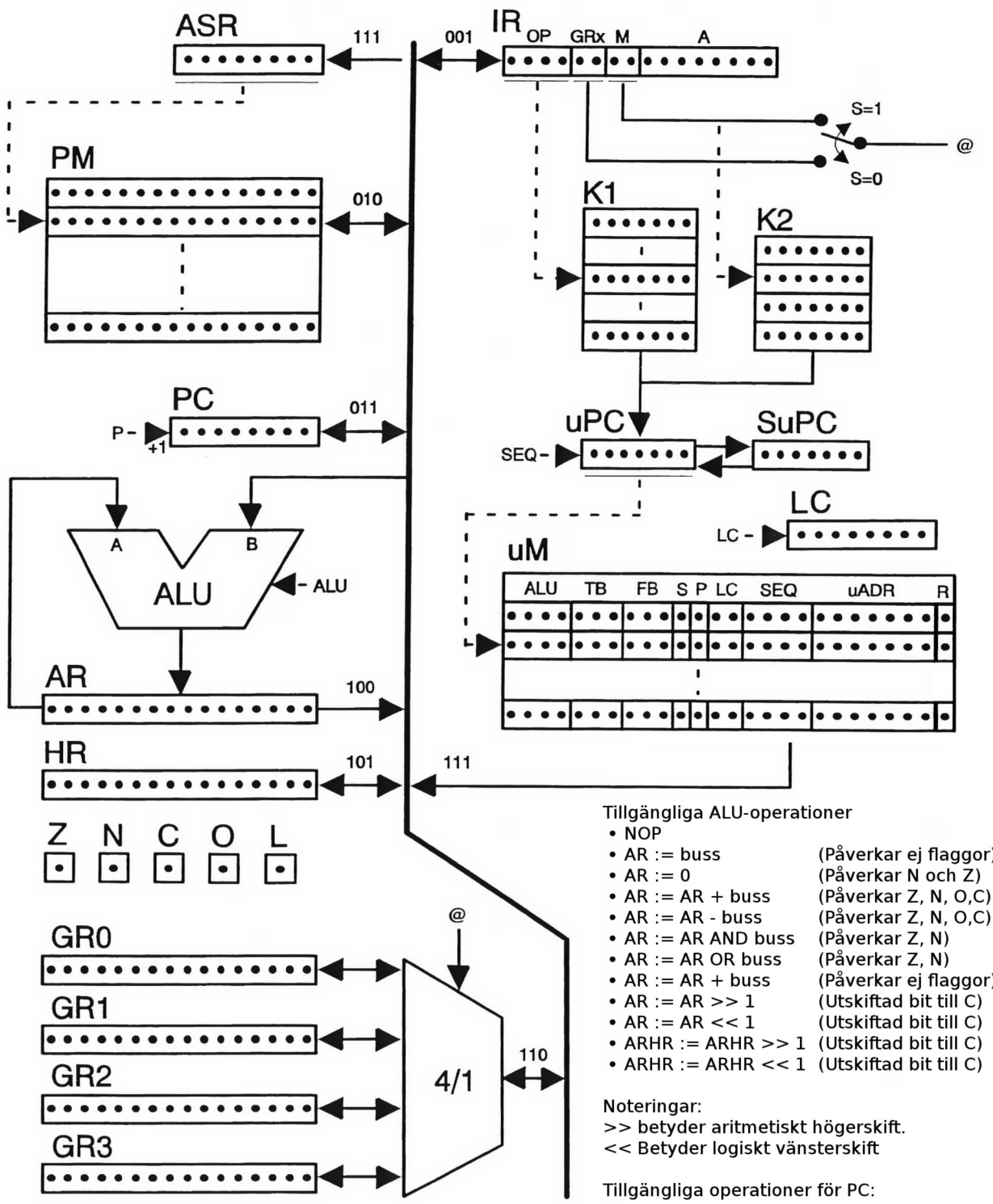
Ange även adresserna för mikrokodsinstruktionerna.

Opcode	Instruktion	Betydelse	Adresseringsmoder	Påverkar flaggor
0001	DECJNZ	GRx := GRx - 1 Hopp PC = A om GRx inte är noll	- (ange 00)	C,Z,N,V

Exempel: PC = $12, GR2 = $37, A = $23. Efter DECJNZ ska GR2 = $36, PC:= $23, Z=0, N=0, V=0,

(b) (2p) Hur många klockcykler (dvs steg) tar det att utföra instruktionen inklusive hämtning av instruktionen. Ange alla olika möjligheter.

(c) (2p) I figur 1 motsvarar varje punkt i registren 1 bit. Hur många adresser (rader) finns det i minnena uM respektive K1?



Tillgängliga ALU-operationer

- NOP
- AR := buss (Påverkar ej flaggor)
- AR := 0 (Påverkar N och Z)
- AR := AR + buss (Påverkar Z, N, O,C)
- AR := AR - buss (Påverkar Z, N, O,C)
- AR := AR AND buss (Påverkar Z, N)
- AR := AR OR buss (Påverkar Z, N)
- AR := AR + buss (Påverkar ej flaggor)
- AR := AR >> 1 (Utskiftad bit till C)
- AR := AR << 1 (Utskiftad bit till C)
- ARHR := ARHR >> 1 (Utskiftad bit till C)
- ARHR := ARHR << 1 (Utskiftad bit till C)

Noteringar:
>> betyder aritmetiskt högerskift.
<< Betyder logiskt vänsterskift

Tillgängliga operationer för PC:

- NOP
- PC := PC + 1 (PC räknas upp med ett)
- PC := buss

Tillgängliga operationer för LC:

- NOP
- LC räknas ned med ett
- LC laddas från uADR

Tillgängliga operationer för uPC:

- uPC := uPC + 1 (uPC räknas upp med ett)
- uPC := K1(OP)
- uPC := K2(M)
- uPC := 0
- uPC := uADR
- uPC := uADR om (valfri) flagga är 1, annars uPC+1
- uPC := uADR om (valfri) flagga är 0, annars uPC+1

Under varje klockcykel kan även en av följande enheter skicka data från bussen: IR, PM, PC, AR, HR, GRx eller uM. En av följande enheter kan också ta emot data ifrån bussen varje klockcykel: IR, PM, PC, AR, HR, GRx eller ASR Viktigt: När uM (de 16 minst signifikanta bitarna) skickas ut till bussen kan enbart en ALU-operation och/eller en bussöverföring göras. uPC räknas också automatiskt upp med ett i detta fall.

Signalen S väljer om M eller GRx-fältet ska användas till att adressera GR0-GR3 (S=0 innebär att GRx-fältet adresserar GR0-GR3). Slutligen används signalen R=1 för att tvinga styrsignalen @ att bli 3.

Figur 1: En välkänd datormodell (Björn Lindskog 1981)

Fråga 2: Allmän teori (10p)

(a) (2p) Ange för och nackdelar med att implementera subrutinanrop med länkregister (som ARM-processorn gör) istället för automatisk stackhantering (som 68000-processorn gör).

(b) (2p) Varför innehåller alla datorer någon form av icke-volatilt minne (ROM, FLASH etc.)?

(c) (2p) Vad betyder egenskapen big-endian hos en processor?

(d) (2p) Beskriv två viktiga funktioner som kan implementeras med hjälp av virtuellt minne.

(e) (2p) Vad innebär delay slot i en processor?

Fråga 3: Assemblerprogrammering (8p)

I minnet ligger en text lagrad som ett antal olika textsträngar utspridda i minnet. I en tabell anges starten till alla textsträngar. Efter sista adressen i tabellen finns 32-bitars värdet 0 som anger att tabellen är slut. Varje textsträng har 1 byte per tecken, och avslutas med en byte med värdet 0.

Skriv en subrutin som räknar fram hur många tecken som texten innehåller. Den sista byten i varje sträng som har värdet 0 ska inte räknas med. Tabellens startadress anges i A0, och totalt antal tecken ska returneras i D0 som ett 32-bitars tal. Förutsätt att adressen i A0 är jämnt delbar med 4 (t ex $0000402C men ej $0000401B).

Exempel: A0 = $00004014, minnet innehåller

Adress	Värde	Adress	Värde
$00004010	$01230044	$00004030	$00004022
$00004014	$00004041	$00004034	$00004020
$00004018	$00004037	$00004038	$686F7070
$0000401C	$00004046	$0000403C	$00000000
$00004020	$00000000	$00004040	$0048656A
$00004024	$01004901	$00004044	$00222100
$00004028	$00222344	$00004048	$12345780
$0000402C	$FCDE1234	$0000404C	$88FF32F2

Efter anrop till subrutinen ska D0 innehålla $00000009 (3 adresser i tabellen, ger 3 tecken på adress $00004041 t o m $00004043, 5 tecken på adress $00004037 t o m $0000403B samt 1 tecken på adress $00004046).

Fråga 4: Aritmetik (6p)

a) (2p) Bestäm det decimala värdet för 5-bitars 2-komplementstalet 11011 samt det decimala värdet för det positiva binära 5-bitarstalet 11011 .

b) (2p) Beräkna additionen av 6-bitars 2-komplementstalet 010010 med 3-bitars 2-komplementstalet 110. Ange svaret i 2-komplementsform.

c) (2p) Betyder en 1:a på flaggan N alltid att en beräkning gav ett negativt svar? Motivera ditt svar.

Fråga 5: Avbrott (8p)

En display med 4 siffror drivs med ett tutor-system. På adress $10080 sitter en 8-bitars port som både kan läsas och skrivas. Bit 7 t o m 4 anger vilket tal som ska finnas på siffran längst till vänster. Bit 3 t o m 0 anger vilket tal som ska finnas på siffran näst längst till vänster. På adress 10082 sitter en 8-bitars port som både kan läsas och skrivas. Bit 7 t o m 4 anger vilket tal som ska finnas på siffran näst längst till höger. Bit 3 t o m 0 anger vilket tal som ska finnas på siffran längst till höger.

Skriv en avbrottsrutin som skiftar siffervärdena åt vänster samt sätter värdet på siffran längst till höger till värdet hos siffran längst till vänster.

Exempel: Om displayen visar siffrorna 1234 ska avbrottsrutinen ändra displayen så siffrorna 2341 visas.

Fråga 6: Cache (8p)

En processor med 32-bitars adress har en cache på 2 Kbyte (2048 bytes, dvs 2^11 bytes). Denna cache är en 2-vägs gruppassociativ cache med cachelinelängd av 32 byte. Ersättningsalgoritmen behåller det värde som lästs senast. Ett bildbehandlingsprogram som körs i processorn läser 32-bitars pixlar minnet på adresserna enligt sekvensen: $10000+ $100*n, dvs $10000, $10100, $10200 etc. Antag att cachen från början är tom.

(a) (2p) Hur många cachelines finns i cacheminnet totalt?

(b) (2p) Hur många bitar av adressen används som tag i cacheminnet?

(c) (2p) Hur många pixlar hinner beräkningsprogrammet läsa innan gamla värden börjar tas bort ur cachen?

(d) (2p) Hur påverkas antal cachemissar om sekvensen startar på adress $1001E istället?

Kort repetition av M68000

Mnemonic	Kort förklaring	Mnemonic	Kort förklaring
ADD	Addition	EXT	Sign extend
ADDX	Add with X flag	EXTB	Sign extend a byte to 32 bit
AND	Logic and	JSR	Jump to subroutine
ASL	Arithmetic shift left	LEA	Load effective address
ASR	Arithmetic shift right	LSL	Logic shift left
BCC	Branch on carry clear	LSR	Logic shift right
BCS	Branch on carry set	MOVE	Move
BEQ	Branch on equal	MULS	Signed multiplication
BGT	Branch on greater than	MULU	Unsigned multiplication
BLT	Branch on less than	NEG	Negate
BNE	Branch on not equal	NOP	No operation
BRA	Branch always	NOT	Bitwise logic invert
BSR	Branch to subroutine	OR	Logic OR
CLR	Clear	ROL	Rotate left
CMP	Compare (Destination - Source)	ROR	Rotate right
DIVS	Signed division	RTE	Return from exception
DIVU	Unsigned division	RTS	Return from subroutine
EOR	Logic XOR	SUB	Subtract
EXG	Exchange	TST	Set integer condition codes

```
; Exempel på M68000 kod som kopierar 200 bytes från $2000 till $3000
    MOVE.L #$2000,A0
    MOVE.L #$3000,A1
    MOVE.B #50,D0
loop
    MOVE.L (A0)+,(A1)+
    ADD.B #-1,D0
    BNE loop
```