

TSEA22 – Föreläsning 1

Introduktion, grindar och kombinationskretsar

Oscar Gustafsson

Kursdeltagare

- Datorteknik, 96 deltagare
- Industriell ekonomi, 19 deltagare
- **Totalt: 115 deltagare**

Se till att ni är registrerade på kursen!

Kursupplägg

- Föreläsningar
 - Distans (tills vidare)
 - Ev. kort introduktion
 - Se på videoinspelningar
 - Ställ frågor (föreläsare finns kvar)
- Lektioner
 - Vanliga salslektioner
 - Problemlösning
 - Två datorlektioner (del av labserien)

Håll er i fas med föreläsningar och lektioner!

Kursupplägg, forts

- Laborationer
 - Laboration 1 och 2: uppkoppling med diskreta kretsar
 - Laboration 3 och 4: beskrivning med VHDL och uppkoppling med programmerbar krets
 - Anmälan öppnar 16:00 idag, nominellt i par
 - Laborationer är typsikt uppdelade i två tillfällen
 - Laborationskompendium för 1 och 2 (strax) tillgängligt
 - Laborationskompendium för 3 och 4 kommer senare
 - Boka gärna in er på alla pass redan nu (så slipper ni glömma bort det)

God förberedelse krävs!

Kursupplägg – tre föreläsare

- Anders Nilsson, examinator (ny för i år)
 - Även examinator i
 - TSEA57/TSEA82 Datorteknik I/D (VT2)
 - TSEA83 Dator konstruktion (D2 VT1+2)
 - TSEA29 Konstruktion med mikrodata (D3 HT1+2)
- Mattias Kryander (examinator –2020)
 - Även examinator i
 - TSEA56 Kandidatprojekt i elektronik (I3 VT1+2)
- Oscar Gustafsson (examinator 2021)
 - Även examinator i
 - TSEA26 Inbygga DSP-processorer (åk 4)
 - TSEA87 Applikationsspecifika integrerade kretsar (åk 4)
 - TSEA84 Digitalt konstruktionsprojekt (åk 5)

Lite om mig

- Läst Y 1992-1998
- Doktorerat 1998-2003
- Docent sedan 2008 (borde fylla i lapparna för att bli professor snart...)
- Avdelningschef på avdelningen för datorteknik
- Handleder för närvarande fem doktorander och två postdoktorer
 - Beräkningsarkitekturer för 5G/6G med fokus på massiv MIMO
 - Beräkningsarkitekturer för optisk fiberkommunikation
 - Flexibla vektorenheter för maskininlärning (och mer)
 - Automatisk generering av beräkningsarkitekturer för simuleringsmodeller
- (Cyklar en del, instruktör på Campushallen)

Kurslitteratur

- Hemert, Digitala kretsar
 - Det mesta av teorin
 - Lektionsuppgifter
- Kurshemsidan/Lisam
 - Föreläsningsmaterial
 - En del lektionsmaterial
 - Labkompendium
 - Extramaterial



Jämfört med förra året

- Två datorlektioner med obligatoriska uppgifter
- Tre tillfällen för laboration 4 (istället för två)

Kursmål

Teori, problemlösning, byggblock, konstruktion, felsökning av digitala system

Digitala system

- Binärkodat
 - In- och utsignaler representeras av
 - 0 (låg spänning)
 - 1 (hög spänning)
- Grindar
 - Bestämmer utsignalens värde baserat på en eller flera insignaler
 - Exempel:
 - Och-grind: utsignalen är 1 endast om alla insignaler är 1 (insignal 1 och insignal 2 och insignal 3 och ...)
 - Eller-grind: utsignalen är 1 om minst en insignal är 1 (insignal 1 eller insignal 2 eller insignal 3 eller ...)
 - Inte (inverterare): utsignalen är värdet som insignalen inte är (inte insignal 1)
 - Exklusivt-eller-grind: utsignalen är 1 om exakt en insignal är 1 (antingen insignal 1 eller insignal 2)

Digitala system

- Boolesk algebra
 - Det finns räkneregler för att kunna uttrycka grindarnas funktioner
 - Exempel:
 - Och: motsvarar “multiplikation”, $y = a \text{ and } b \text{ and } c \Leftrightarrow y = abc$
 - Eller: motsvarar “addition” (som blir max 1), $y = a \text{ or } b \text{ or } c \Leftrightarrow y = a + b + c$
 - Inte: betecknas med ett $'$, t ex $y = a'$ (i viss litteratur med streck ovanför $y = \bar{a}$)
 - Exklusiv eller: exakt en (av två), $y = a \text{ xor } b \Leftrightarrow y = a \oplus b$
- Register/vippor (kommer senare i kursen)
 - Sparar signalers värden
 - Tid introduceras, olika saker kan hända vid olika tidpunkter

Inte bara digitala system

Boolesk algebra används vid t ex programmering, exempelvis

```
^^I^^Iif not ((Y /= 0 and X >= 0) or (Y = 0 and X < 0))
^^I
```

Låt a motsvara $Y \neq 0$ och b motsvara $X \geq 0$, så blir uttrycket

```
^^I^^Iif not ((a and b) or (not a and not b))
^^I
```

vilket kan förenklas som

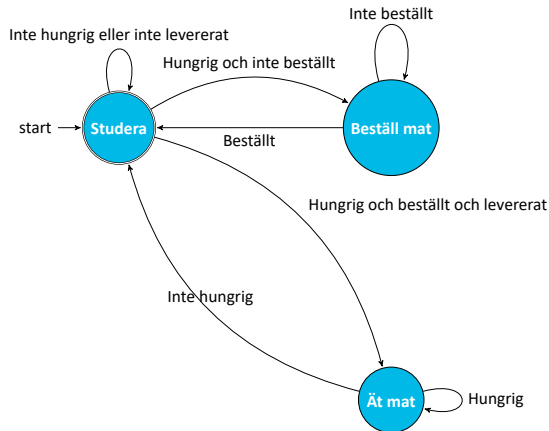
```
^^I^^Iif (a or b) and (not a or not b)
^^I
```

och vidare till

```
^^I^^Iif a xor b
^^T
```

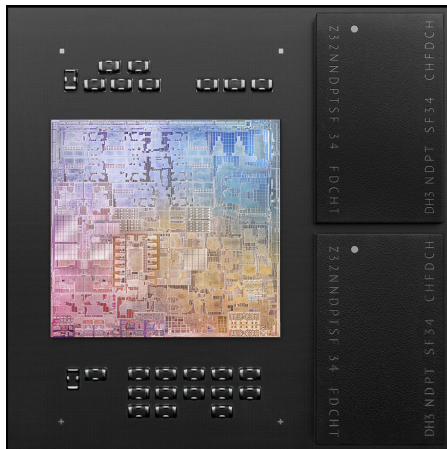
Inte bara digitala system, forts

- Tillståndsdiagram beskriver mer än digitala system



Digitala system kan bli stora

Även om ni inte kommer att bygga så stora system i denna kursen så är det egentligen bara att fortsätta bygga större.



Apple M1, foto:
Apple

Övrigt

- För att få bättre förståelse för grindar och Boolesk algebra kan följande spel(!) rekommenderas
 - Make it True (amerikanska symboler, både Google Play och App Store)
 - logop (grindtyp som text, bara Google Play(?))

www.liu.se