

## **PM inför tentamen i Digital- och datorteknik 2010/2011.**

### **Digitalteknik:**

- \* Omvandla mellan decimala, binära och hexadecimala talsystemen.
- \* Principen för teckenbeloppsrepresentation resp. baskomplementrepresentation av tal med tecken.
- \* Använda binär aritmetik och 2-komplementaritmetik. Kunna subtrahera på en dators vis. Flaggor.
- \* Veta innebörden av naturlig binärkod (NBC), decimala binärkoder (NBCD, m fl) och GRAY-koden.
- \* Känna till principen för flyttal och formatet för främst 32-bitars flyttal, enligt IEEE-standard 754-1985. Känna till hur 0 och  $\infty$  kodas. Känna till undantaget för tal med mycket litet absolutbelopp. Kunna packa upp ett givet flyttal enligt standarden ovan till ett decimalt tal samt den omvända processen.
- \* Läsa och tolka tabeller med ASCII-kod. Veta vad udda och jämn paritet är.
- \* Boolesk algebra. Kunna tillämpa den. Veta vad max- och mintermer är samt SP (disjunktiv) resp PS (konjunktiv) form. Normal form. Principen för karnaughdiagram. Minimera booleska uttryck med karnaughdiagram.
- \* Logikelementen NOT, AND, OR och XOR samt deras inverser, symboler och funktionstabeller.
- \* Analysera kombinatoriska logiknät. Reducera kombinatoriska logiknät till minimal form. Analysera och syntetisera (konstruera) AND/OR-, OR/AND-, NAND- och NOR-logik. Beskriva kombinatoriska nät med grundläggande logiksymboler, booleska uttryck och funktions-tabeller. Konstruera enkla kombinatoriska nät t ex av typen heladderare, kodomvandlare, avkodare, fördelare, väljare (multiplexer). Kunna tolka symboler för kombinatoriska nät i KMP och ARB.
- \* Konstruera kombinatoriska nät med hjälp av väljare och logikgrindar. Konstruera kombinatoriska nät med hjälp av avkodare och logikgrindar. Principen för "three-state"-grindar.
- \* Sekvensnät  
SR-latchen. Klockade latchar av typ SR och D. Vippor av typ SR, D, JK och T. Deras symboler, funktions- och excitationstabeller. Kunna tolka symboler för räknare, register och skiftregister.
- \* Analysera enkla synkrona sekvensnät. Rita upp tillståndstabell, tillståndsgraf, tidsdiagram (pulsdiagram).
- \* Konstruera (syntetisera) autonoma synkrona räknare, synkrona räknare med räknevillkor och små allmänna synkrona sekvensnät, t ex av typ bitsekvensdetektor.
- \* Beskriva funktionen hos de kretsar vars logiksymboler är givna i KMP och ARB.
- \* Beskriva med RTN och styrsignaler de klockcykler som krävs för att utföra olika sammansatta operationer på data i en enkel dataväg enligt figur 7.32 i KMP.

## **Ext-17** (Ver 2011-05-05)

### **Datorteknik:**

- \* Uppbyggnaden av von neumanndatoren. Registeröverföring, RTN, von Neumanns idé, ALU, Styrenhet etc.
- \* FLEX-processorn:  
Samtliga interna register. Hur registren används. Hur stacken fungerar. Statusregistrets flaggor och deras användning. Hur minnesaccesser görs. Hur instruktionsuppsättningen används. Enkla assemblerprogram för FLEX-processorn. Användning av stackpekare SP och indexregister X för adressering. Beskriva och analysera hoppvillkoren för villkorliga hopp. Speciellt skillnaden mellan hoppvillkoren för tal utan och med inbyggt tecken. FLEX-processorns styrenhet.
- \* Ansluta inportar och utportar till FLEX-datorns bussar.
- \* Slå upp instruktioner i instruktionslistan INS1. RTN beskrivning av EXECUTE-sekvenser för instruktioner. Kunna identifiera en given RTN-sekvens som EXECUTE av en maskininstruktion i instruktionslistan. Bestämma antal klockcykler resp bytes för en instruktion.
- \* Använda instruktionslistan INS1. Instruktionsbeskrivningarna och processorns adresseringsmoder. Beräkna antal maskincykler resp bytes som en instruktionssekvens kräver. Beräkna exekveringstider.
- \* Rita och förstå en flödesplan.
- \* Handassemblera och ange programkodens placering (adressen) i minnet.
- \* Analysera resp skriva programavsnitt (inkl subrutiner) i assemblerspråk för FLEX-datorm. Skriva programavsnitt för maskinassemblering med assemblern för FLEX-datorm.
- \* Skriva radkommentarer som förklarar använd algoritm, hoppvillkor, etc för ett program.
- \* Kunna använda och förklara hur ”assembler directives” används och fungerar.
- \* Beskriva metodiken för testning och felsökning av ett FLEX-program med hjälp av simulatorm.

**Instruktionslistan INS1 för FLEX-processorn skall medtagas och användas vid tentamen!**

**Att kunna på digital- och datordelen: Uppgifterna i KMP, ARB och lab-PM!**