

PM inför tentamen i Digital- och datorteknik lp 4, 2013.

Digitalteknik:

- * Omvandla tal mellan decimala, binära och hexadecimala talsystemen.
- * Principen för teckenbeloppsrepresentation och baskomplementrepresentation av tal med tecken.
- * Tillämpa decimal addition och subtraktion med 10-komplementaritmetik.
- * Tillämpa binär addition och subtraktion och 2-komplementaritmetik på heltal. Kunna subtrahera på en dators vis. Flaggsignaler som bildas vid binär addition och subtraktion.
- * Principen för multiplikation och division av binära heltal utan tecken.
- * Veta innebörden av naturlig binärkod (NBC), excess- 2^{n-1} -kod, decimala binärkoder (NBCD, m fl) och GRAY-koden.
- * Känna till principen för flyttal och formatet för främst 32-bitars flyttal, enligt IEEE-standard 754. Känna till hur 0 och ∞ 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) och 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.

Datorteknik:

- * Uppbyggnaden av von neumann datorn. Registeröverföring, RTN, von Neumanns idé, ALU, Styrenhet etc.
- * FLIS-processorn:
Samtliga interna register. Hur registren används. Hur stacken fungerar. Statusregistrets flaggor och deras användning. Hur minnesaccesser görs. Hur instruktionsuppsättningen används. Enkla assemblerprogram för FLIS-processorn. Användning av stackpekare SP och indexregistren X och Y för adressering. Beskriva och analysera hoppvillkoren för villkorliga hopp. Speciellt skillnaden mellan hoppvillkoren för tal utan och med inbyggt tecken. FLIS-processorns styrenhet. Händelseförloppet vid start av processorn (RESET), "TRAP" (ej implementerade OP-koder) och avbrott.
- * Ansluta inportar och utportar till FLISP-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 och bytes för en instruktion.
- * Använda instruktionslistan INS1. Instruktionsbeskrivningarna och processorns adresseringsmoder. Beräkna antal maskincykler och bytes som en instruktionssekvens kräver. Beräkna exekveringstider.
- * Rita och förstå en flödesplan.
- * Handassemblera och ange programkodens placering (adressen) i minnet.
- * Analysera och skriva programavsnitt (inkl subrutiner) i assemblerspråk för FLISP-datorn. Skriva programavsnitt för maskinassemblering med assembleratorn för FLISP-datorn.
- * Kunna beskriva den hårdvara som behövs för att använda avbrottssystemet med en yttre avbrottskälla hos FLISP-dator.
- * Kunna beskriva hur man med hjälp av avbrottssystemet med normal hårdvara och en pulsgenerator kan få FLISP-datorn att köra två olika evighetssnurror pseudoparallellt.
- * 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 FLISP-program med hjälp av simulatoren.
- * Kunna beskriva olika egenskaper hos minnesmoduler som kan placeras i adressrummet hos processorn CPU12.
- * Kunna beskriva hur olika minnesmoduler och in- utportar ansluts till bussarna på CPU12.
- * Kunna rita en "minneskarta" som visar hur minnesmoduler och in-/utportar är placerade i adressrummet hos CPU12 utgående från CS-logik och givna modulstorlekar.
- * Analysera och konstruera kombinatoriska nät för chip-select-avkodning med fullständig (och ofullständig) avkodning. Principen för användning av moduler med "three-state"-utgång.

Den som är inskriven före HT 2012 kan istället för FLIS-processorn använda FLEX-processorn för uppgifter som rör dataväg, styrenhet och assemblerprogrammering för FLISP.

Instruktionslistan (handboken) INS1 för FLIS- (eller FLEX-processorn), skall medtagas och användas vid tentamen!

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