Tentamen i

Digitala System – EDI610 15 hp

Varav denna tentamen 4,5 hp

Institutionen för elektro- och informationsteknik

Campus Helsingborg, LTH

2016-08-22 kl. 14.00 – 19.00

Uppgifterna i tentamen ger totalt 60 poäng. Uppgifterna är inte ordnade på något speciellt sätt.

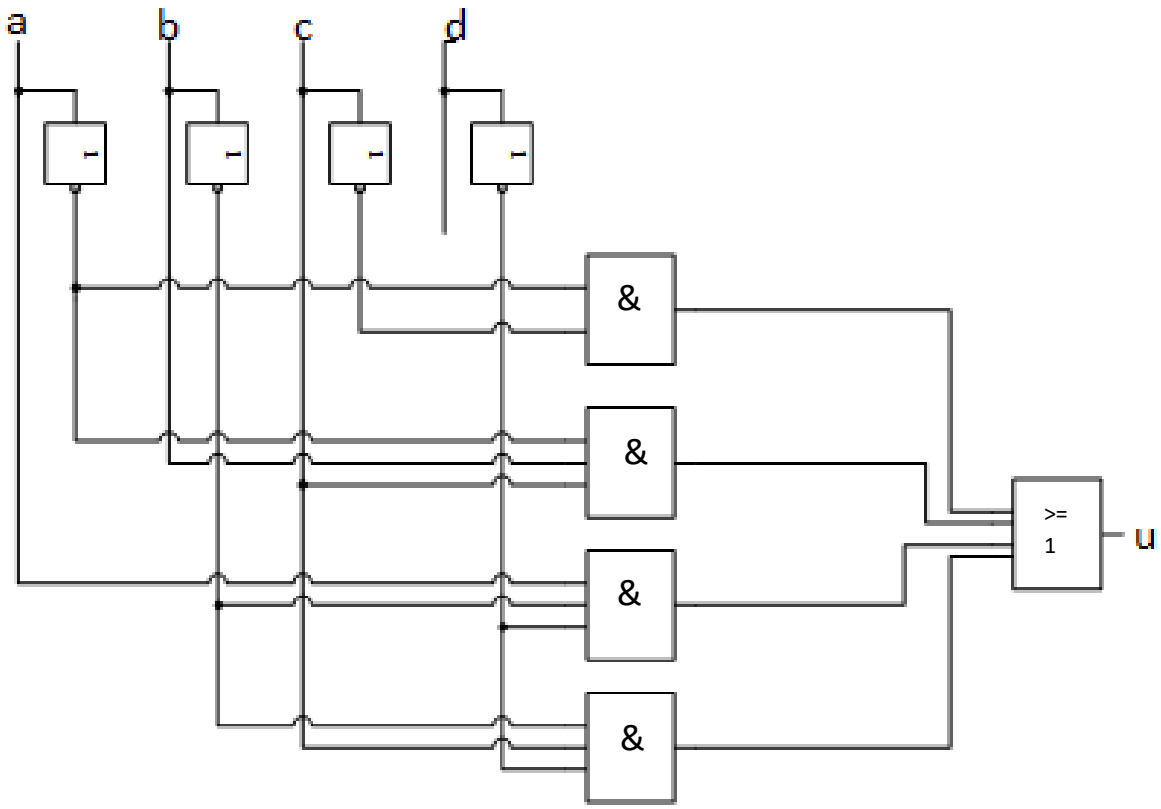
Läs därför igenom alla uppgifter innan du börjar lösa dem. Några uppgifter är uppdelade i deluppgifter. Av totalt 60 möjliga poäng fordras minst 30 för godkänt.

Inga hjälpmedel är tillåtna

- För att rättning av lösning skall komma i fråga fordras att den är **läslig samt klart och tydligt uppställd.**
- Glöm inte att skriva logo och kod på alla blad.
- Alla lösa blad ska vara samlade i omslaget
- Lösningarna ska vara numrerade och ordnade i nummerföljd

Lycka till!

1.



Ovan ser du ett kombinatoriskt nät med fyra ingångar (a,b,c och d) samt en utgång u.

a) Skriv upp det booleska uttrycket för nätet. (**1p)**
b) Skriv sanningstabellen för nätet. **(1p)**
c) Förenkla nätet så mycket det går och rita upp det. **(8p)**

2.

Ett sekvensnät har talen 0 till 15 binärkodade som insignaler (x3; x2; x1; x0). Konstruera ett kombinatoriskt nät som ger tre utsignaler: u0 = 1 om talet är delbart med 2, u1 = 1 om talet är delbart med 3, u2 = 1 om talet är ≤ 9.
OBS! Talet 0 är delbart med alla tal.
Sätt upp sanningstabell, gör ett Karnaughdiagram för vardera utsignal ux och ange dess booleska uttryck. **(10p)**

3.

En tvåbitars upp-räknare med en ingång x, en reset R och en utgång u skall konstrueras. Räknaren skall vara av typ Moore sådan att om x = 0 räknar den binärt och om x = 1 räknar den i Gray-kod. Byte ska kunna ske var som helst i sekvensen. Utsignalen u ska ha värdet u = 1 då räknaren finns i läget {0,0}. Reset, R = 1, ska vara synkron och ställer räknaren i läget {0,0}.
(a) Rita tillståndsdiagram. **(1p)**
(b) Koda tillstånden i vanlig binärkod med starttillståndet 0 och efterföljande som 1, 2, etc. Ställ upp tabellen över möjliga tillstånd q och q+ , samt rita Karnaughdiagrammen. (**4 p)**
(c) Bestäm de minimala q+-funktionerna. **(5p)**

4.

Nedan ser du en VHDL-fil (uppg4_prov) som beskriver ett sekvensnät.

a) Rita tillståndsgrafen för nätet.(**3p**)
b) Skriv sekvensmaskinen på minimal form (SP-form). Koda tillstånden med NBCD-kod. (**7p**)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity uppg4_prov is
    Port ( clock,x,reset : in  STD_LOGIC;
           u : out  STD_LOGIC);
 end uppg4_prov;

architecture Behavioral of uppg4_prov is
type state_type is (s0,s1,s2,s3,s4);
signal present_state, next_state:state_type;
begin
          process(present_state,x,reset)
          begin
           if reset='1' then
              next_state<=s0;
           else
             case present_state is
              when s0 => if x='0' then
                            next_state<=s0;
                              else
                             next_state<=s1;
                         end if;
             when s1 => if x='0' then
                            next_state<=s0;
                              else
                            next_state<=s2;
                         end if;
           when s2 => if x='0' then
                            next_state<=s0;
                              else
                            next_state<=s3;
                        end if;
           when s3 => if x='0' then
                            next_state<=s4; -- ändrat den 5/12 från s3 till s4
                              else
                            next_state<=s3; -- ändrat den 5/12 från s4 till s3
                         end if;
           when s4=> if x='0' then
                            next_state<=s0;
                              else
                            next_state<=s1;
                        end if;
             end case;
          end if;
          end process;

          process(present_state, x)  -- ändrat den 5/12: x tillagd i känslighetsvariablerna
          begin
          if present_state=s4 then
           if x='0' then
                    u<='1';
               else
                    u<='0';
            end if;
          else
                 u<='0';
           end if;
           end process;

          process(clock)
          begin
          if rising_edge(clock) then
          present_state<=next_state;
          end if;
          end process;
end  architecture Behavioral;
```

5.

Antag en processor där instruktioner anges i ett 3-adress format. För t ex ALU operationer anges instruktioner enligt: Operation (OP) Destination (DES), Operand1 (OP1), Operand2 (OP2) och utförs enligt följande: DES = OP1 OP OP2. Om assembler instruktionen är ADD R2, R1, 5 kommer detta att utföras: R2=R1 ADD 5, d v s talet 5 kommer adderas till innehållet i register 1 (R1) och resultatet lagras i register 2 (R2). (**10p**)

För programmet som är givet nedan, besvara:

1. Hur många bitar behövs för att lagra en instruktion?
2. Hur många bitar innehåller instruktionsregistret?
3. Vilka instruktioner kommer exekvera?
4. Vad som händer vid varje instruktion?
5. Vad innehåll i programräknare och instruktionsregister vid varje instruktion som exekverats?

ADRESS	INSTRUKTION	FÖRKLARING
00000	XOR R1, R1, R1	R1<-R1 XOR R1
00010	ADD R2, R1, 5	R2<-R1 + 5
00100	MUL R2, R1, R2	R2<-R1*R2
00110	BEZ 10000	IF ZERO BRANCH TO 10000
01000	SUB R2, R2, 1	R2<-R2-1
01010	BR 10000	BRANCH TO 01110
01100	ADD R1, R1, 1	R1 <- R1 + 1
01110	MUL R1, 5, 2	R1 <- 5 MUL 2
10000	AND R1, R1, R2	R1 <- R1 AND R2

6.

Antag ett datorsystem med ett cacheminne som består av 16 bytes och ett primärminne som består av 64 bytes. Cacheminnet är direktmappat med cacherader om 4 bytes. (**10p**)

1. Hur många bitar behövs för att specificera datorsystemets adressrymd?
2. Vid exekvering av ett program har processorn läst adress 1 och vill läsa på adress 5, är det möjligt att avgöra om den läsningen kommer ge en hit eller en miss?
3. I cacheminnet, när processorn läser i minnet, vilka adresser kan hamna i den första cacheraden?
4. Lokalitet av referenser är en viktig princip för cacheminnen, vad säger principen?
5. Enligt principen om lokalitet av referenser, kan man säga något om storleken av block?