

Datum: 2012-08-29 (5 timmar)

Tillåtna hjälpmedel: Inga

Kursansvarig: Mikael Bylund, tel: 060-148748, e-post: mikael.bylund@miun.se

Antal uppgifter: 9

Antal sidor: 3

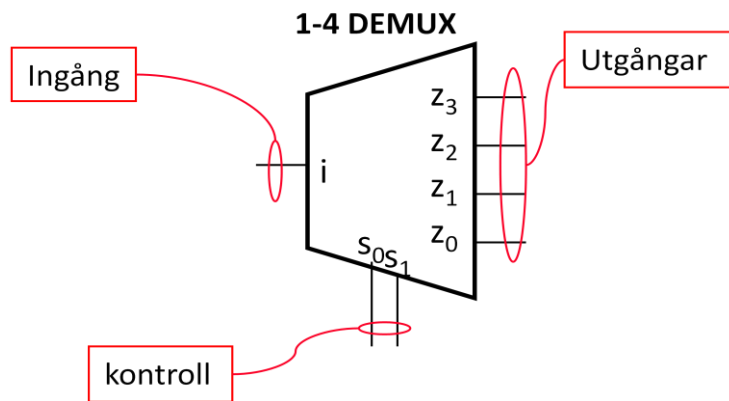
Maxpoäng: 100 (50 poäng krävs för godkänt)

Anvisningar för inlämnade lösningar:

- Resonemang och motiveringar får ej vara så knapphändiga att de blir svåra att följa.
- Tankegången bakom uppställda ekvationer skall förklaras.
- Uträkningarna skall vara tillräckligt fullständiga för att visa hur slutresultatet har erhållits.
- Varje problemlösning skall avslutas med ett klart formulerat svar.

1. **Konvertera det oktala talet 7361 till ett binärt tal. Konvertera sedan detta binära tal till ett hexadecimalt tal. (8p)**
2. **Konvertera det decimala talet -51 till ett 8-bitars binärt tal på 2-komplementform. (8p)**
3. **Minimera följande Booleska uttryck:  $(a + b)(\bar{a} + \bar{b}) + \overline{a \cdot b}$  (10p)**
4. **Avgör om följande likhet gäller:  $a \cdot b + \overline{b \cdot c} + \bar{a} \cdot c = a + \overline{b + c}$  (10p)**
5. **Minimera funktionen  $f(x_3, x_2, x_1, x_0) = \sum (0,1,5,10,11,15) + d(4,7,8,14)$  genom att använda Karnaugh-diagram. Skriv funktionen som ett minimalt Summa-av-Produkt (SP) uttryck. (10p)**
6. **Designa en funktion som tar ett 3-bitars binärt tal som input, adderar ett till detta tal, och sedan ger resultatet som ett 4-bitars binärt tal på utgången. Svara med Booleska uttryck för outputbitarna. (12p)**

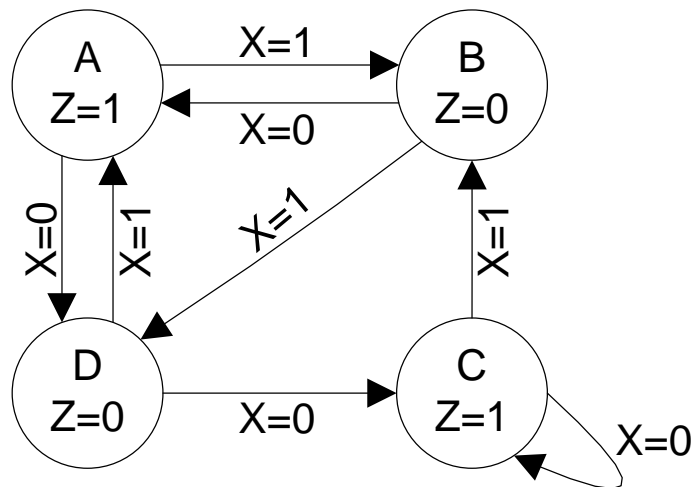
## 7. Skriv VHDL-kod för en 1-4 De-Multiplexer. (12p)



Figur 1

## 8. Rita tillståndsdigrammet för VHDL-koden i Appendix 1. Är detta en Moore- eller en Mealy-maskin? Motivera ditt svar. (14p)

## 9. Utveckla ett kretsschema för tillståndsdigrammet nedan. Använd Graykodning och D-vippor för tillståndsmminnet. (16p)



Figur 2

**APPENDIX 1.**

```
entity seq_design is
port(      a, clock, reset: in std_logic;
        x: out std_logic
);
end seq_design

architecture FSM of seq_design is
    type state_type is (S0, S1, S2, S3);
    signal next_state, current_state: state_type;

    process(clock, reset)
    begin
        if (reset='1') then
            current_state <= S0;

        else
            case current_state is

                when S0 => if a='0' then
                            next_state <= S2;
                            x <= 1;
                        elsif a='1' then
                            next_state <= S1;
                            x <= 0;
                        end if;

                when S1 => if a='0' then
                            next_state <= S2;
                            x <= 1;
                        elsif a='1' then
                            next_state <= S1;
                            x <= 0;
                        end if;

                when S2 => if a='0' then
                            next_state <= S3;
                            x <= 1;
                        elsif a='1' then
                            next_state <= S2;
                            x <= 0;
                        end if;

                when S3 => if a='0' then
                            next_state <= S3;
                            x <= 0;
                        elsif a='1' then
                            next_state <= S0;
                            x <= 1;
                        end if;

            end case;
        endif;

        current_state <= next_state;

    end process;
end FSM;
```