

1

timer (0) : 10/2 MHz

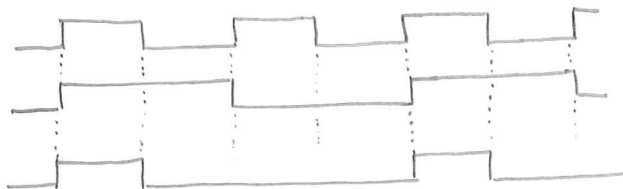
timer (1) : 10/4 MHz

...

timer (5) : 10/2⁶ MHz = 156 kHz

timer (6) : 10/2⁷ MHz = 178 kHz

timer (5) AND timer (6):



(a)-(b) Enligt skiss ovan har pulse = timer(5) AND timer(6) samma frekvens som timer(6), dvs 178 kHz, och duty cycle 1/4.

(c) signal timer: std_logic_vector (7 downto 0);
...
pulse <= timer (6) and timer (7);

(d) Signalen skall ha en period på 1/30 kHz = 33.3 μs, dvs 333 klockpulser, varav den är hög under 111 klockpulser.

1 architecture ändras:

```
signal timer : std_logic_vector (8 downto 0);
```

```
signal y : std_logic;
```

Efter if rising-edge (clock) then:

```
timer <= timer + 1;
```

```
if timer = 110 then
```

```
  y <= '0';
```

```
elsif timer = 332 then
```

```
  timer <= (others => '0');
```

```
  y <= '1';
```

```
end if;
```

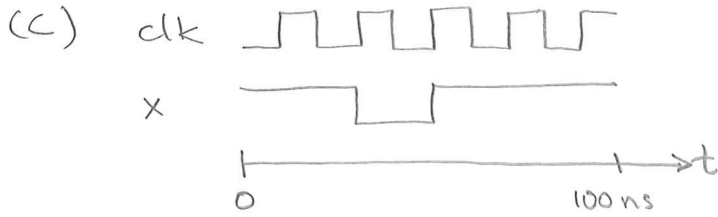
Och pulse-raden ändras till:

```
pulse <= y;
```

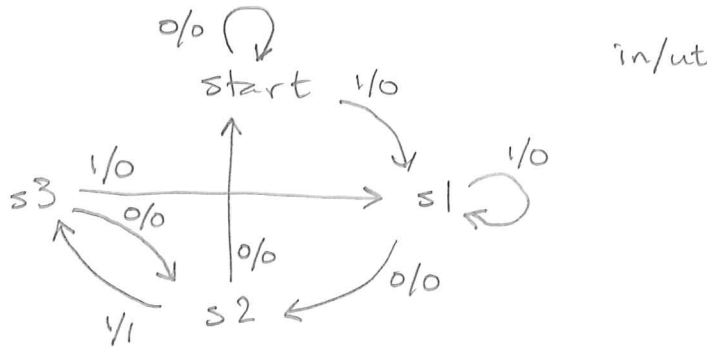
- Många andra lösningar är möjliga.
- Egentligen behövs en reset också, fast det ingick inte i uppgiften.

2. (a) do-fil

(b) run 100 ns (eller bara run 100)



3. (a)



(b) ENTITY:

reset: IN std_logic;

Alternativ 1, i första processen:

```

PROCESS (x, s, reset)
BEGIN
  IF reset = '0' THEN
    nxt <= start;
  ELSE
    nxt <= s;
  END IF;

```

(Använd ett av alternativen)

Alternativ 2, i andra processen:

```

BEGIN
  IF reset = '0' THEN
    s <= start;
  ELSIF rising-edge ...
  ...

```

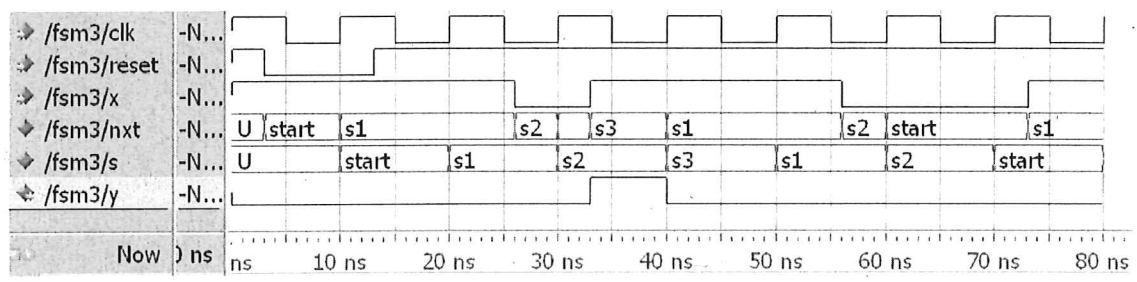
Obs: Det fungerar inte att reseta s i första processen eller nxt i andra. Ingen signal får tilldelas värden i mer än en process!

- 3(c)
- Tillståndet ändras vid klockpuls.
 - Utsignalen ändras när insignalen ändras, även om det inte kommer någon klockpuls då.

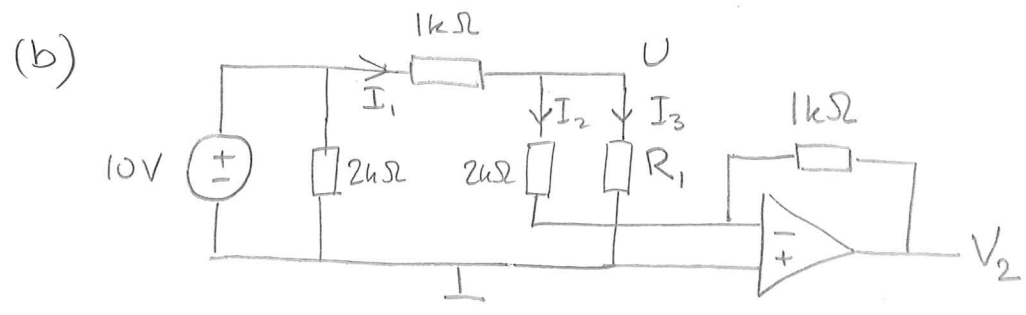
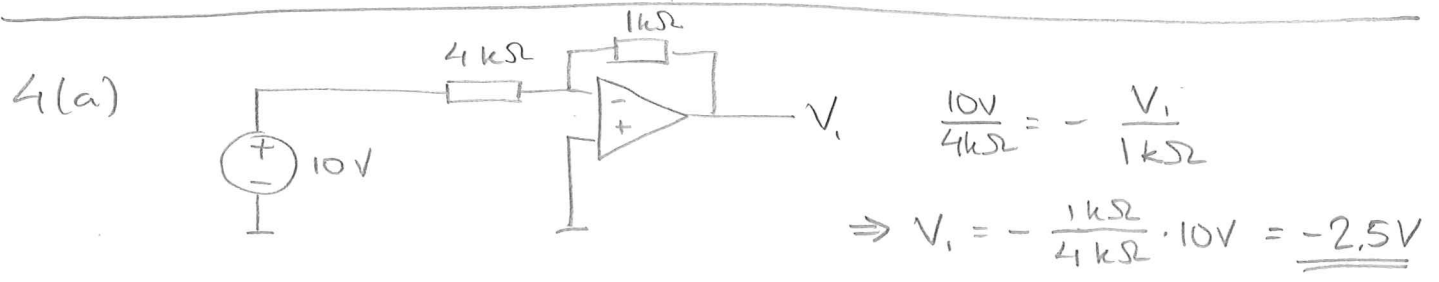
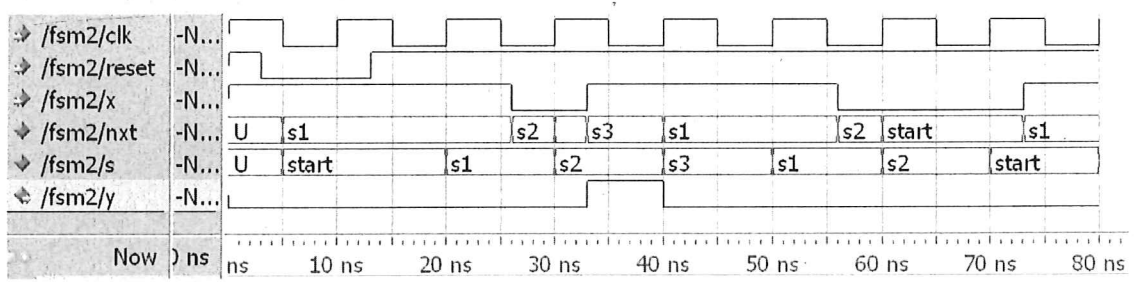
⇒ Mealy

(d) Signalerna skiljer sig en aning beroende på hur reseten implementeras i uppgift (b).

Alternativ 1:



Alternativ 2:



I schemat ovan representerar R_1 fem motstånd enligt

$$R_1 = 1k + 2k // (1 + 2k // 2k) = 2k\Omega$$

Med beteckningarna U, I_1, I_2, I_3 enligt figuren gäller:

$$I_2 = \frac{U}{2k\Omega}$$

$$I_3 = \frac{U}{2k\Omega}$$

$$I_1 = \frac{10V - U}{1k\Omega}$$

$$I_1 = I_2 + I_3 \Rightarrow \frac{10V - U}{1k\Omega} = \frac{U}{2k\Omega} + \frac{U}{2k\Omega}$$

$$\Rightarrow 10V - U = U$$

$$\Rightarrow U = 5V$$

$$\Rightarrow I_2 = 2,5 \text{ mA}$$

$$\Rightarrow V_2 = -I_2 \cdot 1k\Omega = \underline{\underline{-2,5V}}$$

(c) D/A-omvandling

(d) Den andra kopplingen ger noggrannare D/A-omvandling, eftersom motståndens bara har två värden (eller ett enda om man seriekopplar $1k\Omega + 1k\Omega$ för att få $2k\Omega$). Det är lätt att massproducera likadana motstånd i integrerad krets-teknologi och onoggrannheter i resistansen påverkar resultatet mindre. Se Bengtsson avsnitt 4.2.

5. (a) Inimpedans $R_1 + R_2 = \underline{\underline{550k\Omega}}$

(b) Spänningsförstärkning

$$A = \frac{50}{50+500} \cdot \left(-\frac{47}{10}\right) \cdot \frac{16}{16+47} = \underline{\underline{-0,11 \text{ gånger}}}$$

spänningsdelning
inverterande förstärkare
spänningsdelning

(c) Vin har amplitud 5V

\Rightarrow Out har amplitud $A \cdot 5V = 0,54V$ och effektivvärde $\frac{0,54V}{\sqrt{2}} = 0,38V$.

Hörlurens effekt är $\frac{0,38^2}{16} = 9,2 \text{ mW}$, vilket

motsvarar $10 \log \frac{9,2 \text{ mW}}{1 \text{ mW}} = 9,6 \text{ dBm}$.

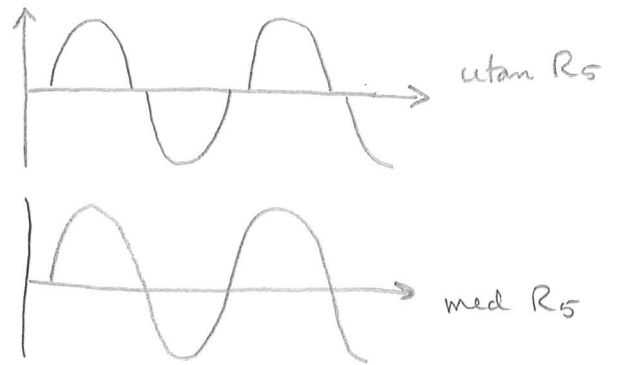
Eftersom $1 \text{ mW} = 0 \text{ dBm}$ motsvarar 100 dB SPL

gäller att $9,6 \text{ dBm}$ motsvarar $100 + 9,6 = \underline{\underline{109,6 \text{ dB SPL}}}$.

5 (d) Spänningsföljaren isolerar spänningsdelningen från förstärkersteget. Tar man bort den så påverkas som är mycket större än strömmen genom R_3 , och förstärkningen minskar kraftigt.
 (Enligt Spice-simulering minskar förstärkningen från 4.7 (13.4 dB) till 0.077 (-22.3 dB).)

(e) Transistorsteget levererar effekt. Operationsförstärkaren ger inte tillräcklig ström för att driva en 16 Ω -hörlur.

(f) R_5 minskar övergångsdistortionen när effektsteget (av klass B) växlar från att ena transistorn leder till den andra, dvs när inspänningen passerar noll.



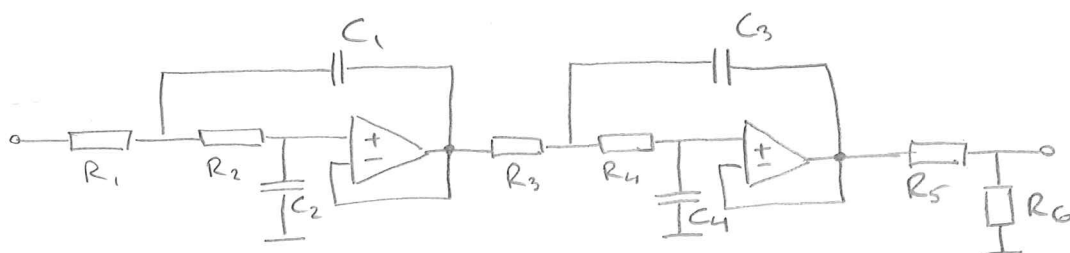
6. Från tabell:

$$P(a) = 1.414(1 + 0.188a + 1.108a^2)(1 + 2.096a + 5.108a^2)$$

$$\omega_0 = 2\pi \cdot 10k = 62.8 \text{ kHz}$$

$$H(s) = \frac{1}{P(s/\omega_0)} = \frac{0.707}{(1 + 2.99 \cdot 10^{-6}s + 2.81 \cdot 10^{-10}s^2)(1 + 3.34 \cdot 10^{-5}s + 1.29 \cdot 10^{-9}s^2)}$$

Dämpningen kan implementeras som en spänningsdelning eller återkopplad operationsförstärkare, medom de båda andragradspolynomen implementeras som Sallen-Key-länkar:



Komponenterna skall uppfylla:

$$(R_1 + R_2)C_2 = 2,99 \cdot 10^{-6}$$

$$(R_3 + R_4)C_4 = 3,34 \cdot 10^{-5}$$

$$R_1 R_2 C_1 C_2 = 2,81 \cdot 10^{-10}$$

$$R_3 R_4 C_3 C_4 = 1,29 \cdot 10^{-9}$$

$$\frac{R_6}{R_5 + R_6} = 0,707$$

Det finns många sätt att uppfylla ekvationerna, t.ex.:

$$R_1 = 3,9 \text{ k}\Omega$$

$$R_3 = 3,3 \text{ k}\Omega$$

$$R_2 = 10 \text{ k}\Omega$$

$$R_4 = 12 \text{ k}\Omega$$

$$R_5 = 3,3 \text{ k}\Omega$$

$$C_1 = 33 \text{ nF}$$

$$C_3 = 15 \text{ nF}$$

$$R_6 = 8,2 \text{ k}\Omega$$

$$C_2 = 0,22 \text{ nF}$$

$$C_4 = 2,2 \text{ nF}$$

7(a) Lägsta samplingsfrekvens: $2,15 \text{ kHz} = 30 \text{ kHz}$

$$14 \text{ bit/sampel} \Rightarrow 14 \cdot 30 = \underline{\underline{420 \text{ kbit/s}}}$$

(b) Syftet är att möjliggöra bit- och ordsynkronisering genom att mottagaren identifierar den flank (stopp \rightarrow start) som förekommer mellan alla dataord, oavsett datainnehåll.

(c) Fortfarande 30 kHz sampling men $18 \cdot 30 = 540 \text{ kbit/s}$

- SAR-en behöver 4 extra tillstånd och 4 extra bitar i parallellporten till D/A-omvandlaren.
- Oscillatorn behöver generera en snabbare klocksignal för transmission av bitarna.
- D/A-omvandlaren behöver bytas mot en med 16 bitar och tillräcklig bandbredd.
- Komparatorns bandbredd (slew rate) behöver kontrolleras mot den högre hastigheten.

(d) Nu 40 kHz sampling och $14 \cdot 40 = 560 \text{ kbit/s}$.

- LP-filtret behöver högre gränshfrekvens
- Oscillatorn behöver generera en snabbare klocksignal för konvertering och transmission.
- D/A-omvandlarens, komparatorns, och mikrofonförstärkarens bandbredder behöver kontrolleras så att de klarar den högre hastigheten.