



# Terning simulator

Tittel: Terning simulator

Forfattere: Andreas Lindeman

Versjon: 2.0

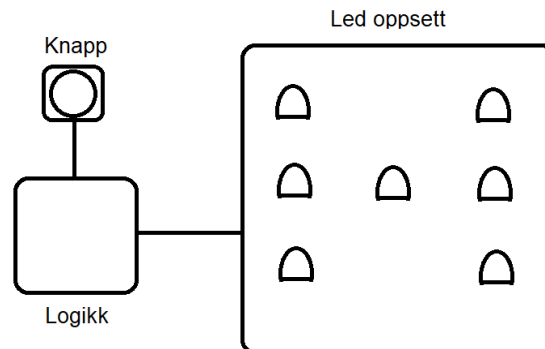
Dato: 15.05.2025

## Innhold

<b>1</b>	<b>Problembeskrivelse</b>	<b>2</b>
1.1	Systemets oppgave og krav til oppførsel . . . . .	2
<b>2</b>	<b>Prinsipiell løsning</b>	<b>3</b>
2.1	Systembeskrivelse . . . . .	3
2.2	Logikk . . . . .	3
2.2.1	Klokkestyring . . . . .	3
2.2.2	Tellerdesign . . . . .	3
2.2.3	LED-styring . . . . .	4
2.2.4	LED Logikk . . . . .	6
2.3	Kretsdesign . . . . .	6
2.4	Beregning av seriemotstander . . . . .	7
2.5	Total effektforbruk . . . . .	7
<b>3</b>	<b>Realisering og test</b>	<b>8</b>
3.1	Logikk . . . . .	8
3.2	Beregning av seriemotstander . . . . .	10
3.2.1	Rød LED . . . . .	10
3.2.2	Orange LED-er . . . . .	10
3.2.3	Gule LED-er . . . . .	11
3.3	Total effektforbruk . . . . .	11
3.3.1	Rød LED . . . . .	11
3.3.2	Orange LED-er . . . . .	12
3.3.3	Gule LED-er . . . . .	12
3.4	Effektforbruk for forskjellige terningkast . . . . .	12
3.5	Forventet effektforbruk . . . . .	12
3.6	Implementering . . . . .	12
3.7	Kretsdesign . . . . .	12
3.8	Testprosedyre . . . . .	13
3.9	Testresultater . . . . .	13
3.10	Sammenligning av teoretisk og realisert effekt . . . . .	14
3.11	Diskusjon . . . . .	14
3.12	Komplett system . . . . .	16
3.13	Fremtidige forbedringer . . . . .	16
<b>4</b>	<b>Konklusjon</b>	<b>17</b>

# 1 Problembeskrivelse

Vi vil ta for oss design av et system som vist i Figur 1.



**Figur 1:** Figur av systemet

Vi vil ta for oss design av et system som simulerer en elektronisk terning ved hjelp av en FPGA. Systemet består av syv lysdioder (LEDs) arrangert i et terningmønster og en trykknapp. Når brukeren trykker på knappen, skal systemet generere et tilfeldig tall mellom 1 og 6, og dette tallet skal vises ved å lyse opp de aktuelle lysdiodene som tilsvarer antall prikker på en vanlig terning.

## 1.1 Systemets oppgave og krav til oppførsel

- **Funksjonelle krav:**

- Systemet skal ha en trykknapp som brukeren kan trykke på for å kasteterningen.
- Når knappen trykkes ned, skal systemet raskt veksle mellom alle mulige terningverdier (1 til 6).
- Når knappen slippes, skal systemet vise terningens verdi.
- Lysdiodene skal være arrangert i et mønster som tilsvarer en vanlig terning.

- **Ikke-funksjonelle krav:**

- Systemet skal bruke en FPGA for å generere de tilfeldige tallene og styre lysdiodene.
- Lysdiodene skal ha forskjellige farger (minst tre ulike farger) og forskjellig effektforbruk.
- Systemet skal ha et lavt effektforbruk, både når knappen holdes nede og når resultatet vises.
- Systemet skal være enkelt å realisere med tilgjengelige komponenter og teknologier.

## 2 Prinsipiell løsning

### 2.1 Systembeskrivelse

Systemet består av en FPGA-brikke (Lattice ICE40) som styrer syv lysdioder (LED-er) arrangert i et terningmønster. Når en trykknapp aktiveres, genererer FPGA-en et tilfeldig tall mellom 1 og 6, og de tilsvarende LED-ene lyser opp for å vise resultatet.

### 2.2 Logikk

For å realisere den tilfeldige oppførselen til terningen, brukes en rask teller som genererer en sekvens av tall fra 1 til 6. Telleren er koblet til den interne klokken til FPGA-en, og den teller så raskt at det er umulig for en bruker å forutsi hvilket tall som vil vises når knappen slippes. Dette gir en illusjon av tilfeldighet.

#### 2.2.1 Klokkestyring

For å styre at klokken bare påvirker systemet når knappen holdes nede brukes følgende logikk.

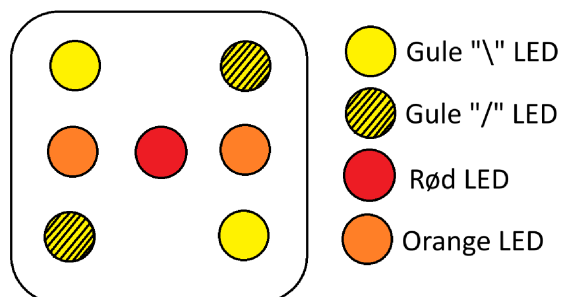
$$Ut = (CLK + \overline{Knapp}) \oplus \overline{Knapp} \quad (1)$$

Hvor  $CLK$  er et klokkesignal som bytter mellom høy og lav i ekstremt høy hastighet, og  $knapp$  samsvarer med statusen til en knapp. Dette medfører at utgangssignalet  $Ut$  normalt er lavt, men følger  $CLK$ -signalet når knappen holdes nede. Dette signalet kan så gå videre til en teller som vil telle et steg for hvert klokkesignal. De raske pulsene som naturlig kommer fra  $CLK$  vil gjøre at telleren teller i ekstrem hastighet når knappen holdes nede, og står stille når knappen er i sin naturlige posisjon.

#### 2.2.2 Tellerdesign

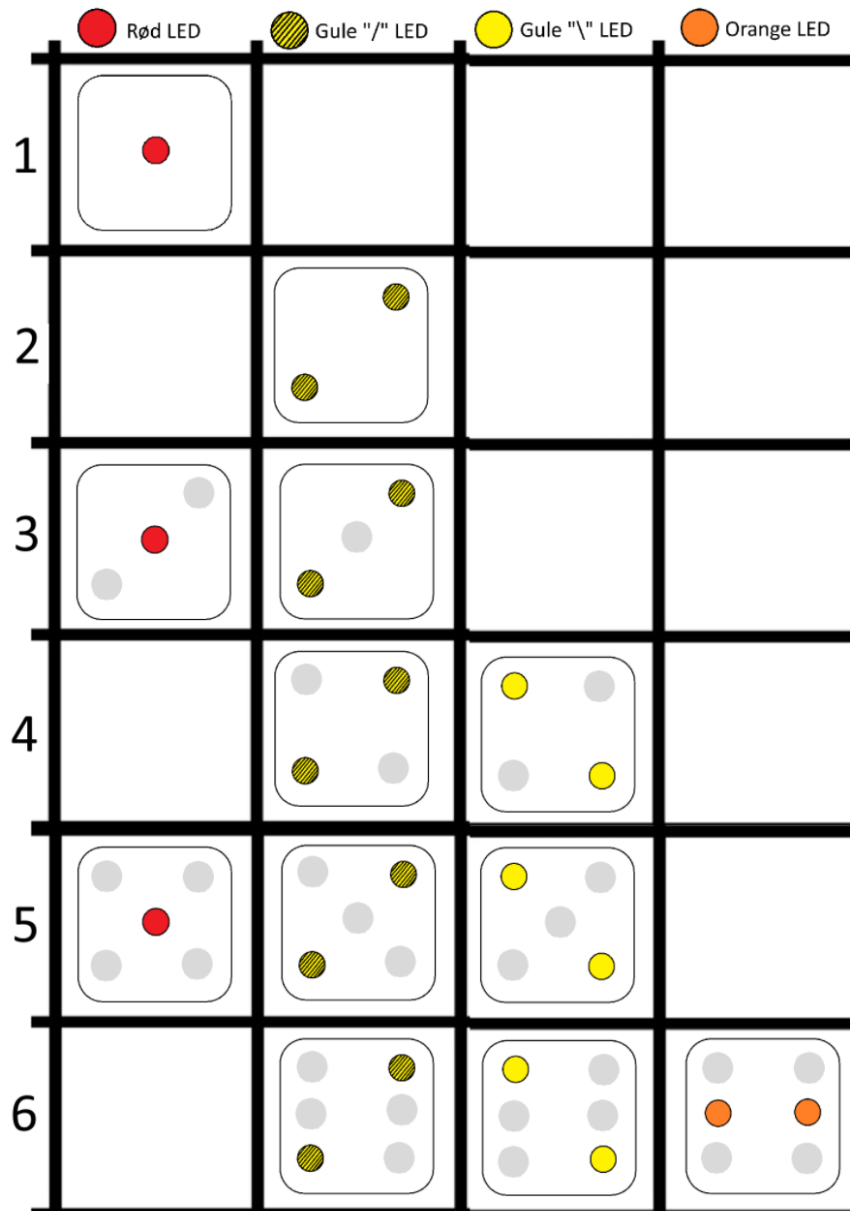
Telleren er en 3-bit binær teller som teller fra 1 til 6 før den begynner på 1 igjen. Den har tre utganger som representerer de binære verdiene 1 ( $A$ ), 2 ( $B$ ) og 4 ( $C$ ). Disse utgangene er koblet direkte til LED'ene for å vise brukeren den nåværende verdien på en forstårlig måte.

### 2.2.3 LED-styring



**Figur 2:** Indeling av LED'ene

LED-ene er gruppert i fire grupper basert på deres posisjon på terningen. I Figur 2 kan du se de fire forskjellige gruppene lysdiodene er delt inn i. I midten har vi en rød LED, på begge sider av denne har vi oransje LED. Vi har så to set med gule LED. Ledene øverst til høyre og nederst til venstre kaller vi Gule "/". De to siste LED'ene er de øverst til venstre og nederst til høyre disse kaller vi de Gule "\" LED'ene. Under ser du et diagram over hvilke LED's som er tent for tilsvarende terning verdier.



Figur 3: Diagram over samholdet mellom terning verdi og LED'er

Fra Figur 3 er det lett og se når de forskjellige LED'ene skal være tent og for hvilke verdier.

- **Rød LED (midt):** Tennes når telleren viser 1, 3, eller 5.
- **Gule "/" LED-er:** Led lysene øverst til høyre og nederst til venstre, tenner på verdien over og inkludert 2.
- **Gule "\" LED-er:** Led lysene øverst til venstre og nederst til høyre, tenner på verdien over og inkludert 4.
- **Orange LED-er:** Tennes kun når terningen viser 6.

## 2.2.4 LED Logikk

For å implementere denne logikken, brukes følgende logiske uttrykk basert på tellerens utganger ( $C$ ,  $B$ ,  $A$ ).

Tall	C	B	A	Rød	Gul "/"	Gul "\"	Orange
1	0	0	1	1	0	0	0
2	0	1	0	0	1	0	0
3	0	1	1	1	1	0	0
4	1	0	0	0	1	1	0
5	1	0	1	1	1	1	0
6	1	1	0	0	1	1	1

**Figur 4:** Sannhetstabell for LED-logikk

I sannhetstabellen over ser du når de forskjellige LED'ene skal lyse for hvilke tellerverdier. Ved bruk av denne tabellen kan vi etablere simpel binær logikk for de forskjellige LED'ene.

$$\text{Rød} = A \tag{2}$$

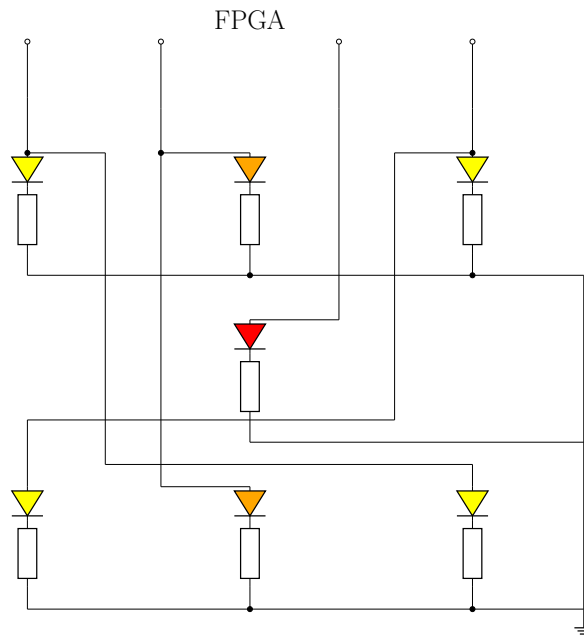
$$\text{Gul "/" } = B + C \tag{3}$$

$$\text{Gul "\" } = C \tag{4}$$

$$\text{Orange} = B \cdot C \tag{5}$$

## 2.3 Kretsdesign

LED'ene som alltid er lyser samtidig er koblet sammen i parallellkoblinger. Den røde LED-en er koblet direkte til FPGA-en, mens de orange og gule LED-ene er koblet i parallelle par for å redusere antall utgangsporter som kreves fra FPGA-en og forenkle systemet. LED-ene er koblet til FPGA-en gjennom seriemotstander for å begrense strømmen og beskytte både LED-ene og FPGA-en.



**Figur 5:** Krettsdiagram av systemet

I Figur 5 ser du et krettskjema om hvordan systemet tenkelig er koblet opp.

## 2.4 Beregning av seriemotstander

For å beregne de nødvendige seriemotstandene, bruker vi Ohms lov og formelen for strømbe-  
grensning i en LED-krets:

$$R = \frac{V_{CC} - V_{LED}}{I_{LED}} \quad (6)$$

Hvor:

- $R$  er motstand [ $\Omega$ ]
- $V_{CC}$  er spenningen fra FPGA-en [V].
- $V_{LED}$  er fremoverspenningen til LED-en [V].
- $I_{LED}$  er ønsket strøm gjennom LED-en [A] (typisk 20 mA for standard LED-er).

## 2.5 Total effektforbruk

Effektforbruket til hver LED kan beregnes ved hjelp av formelen:

$$P[W] = V_{LED} \cdot I_{LED} \quad (7)$$

Forventet effektforbruk av systemet kan så beregnes som gjennomsnittet av effektforbruket for alle seks utfall:

$$P_{\text{forventet}} = \frac{1}{6} \sum_{k=1}^6 \frac{1}{N_k} \sum_{i=1}^{N_k} P_i \quad (8)$$

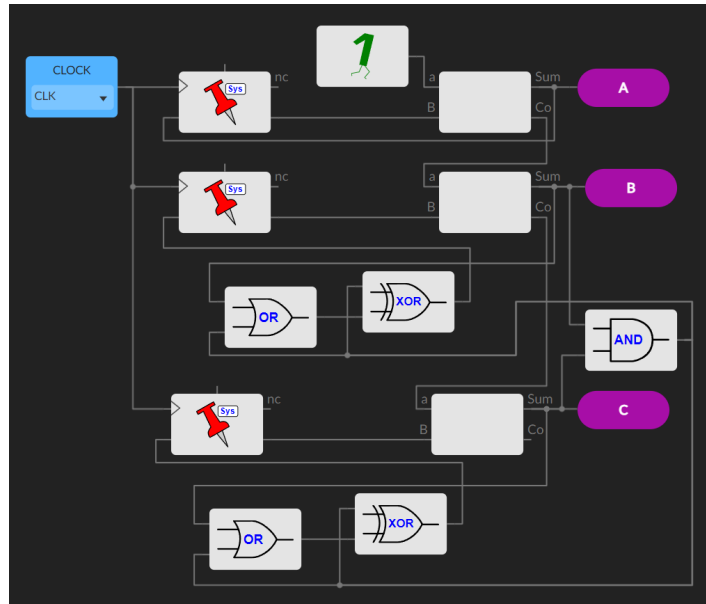
Hvor:

- $P_{\text{forventet}}$  er effekt  $[W]$ .
- $k$  er verdien til terningen.
- $N_k$  er hvor mange LED'er som lyser i terningverdien  $k$ .
- $P_i$  er effektforbruket for en LED som lyser i oppsettet  $N_k$   $[W]$ .

### 3 Realisering og test

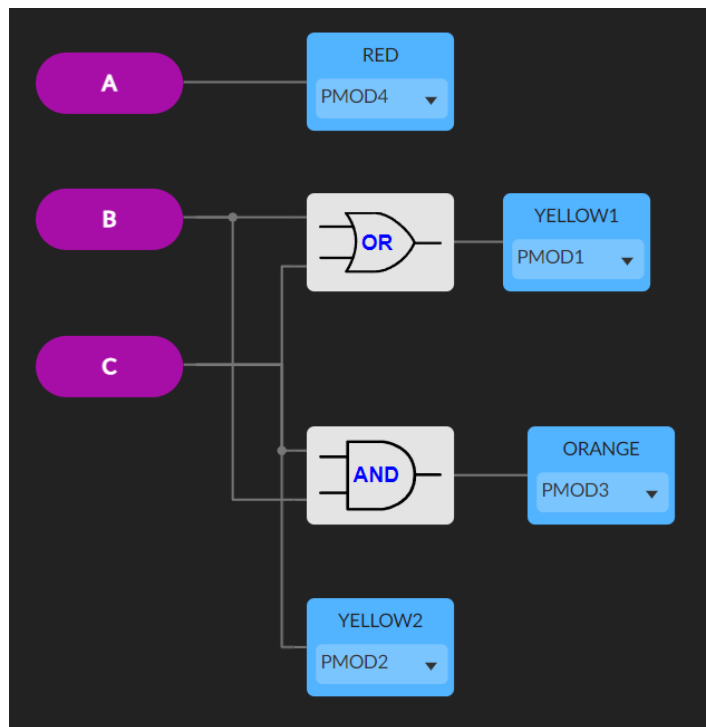
#### 3.1 Logikk

I dette prosjektet er all implementasjon av logikken – inkludert knapplogikk med klokkestyring, tellerdesign og LED-styring – gjort i IceStudio. IceStudio er et grafisk utviklingsmiljø for FPGA-programmering, hvor man bygger logiske kretser ved hjelp av blokker og forbindelser i stedet for å skrive kode.



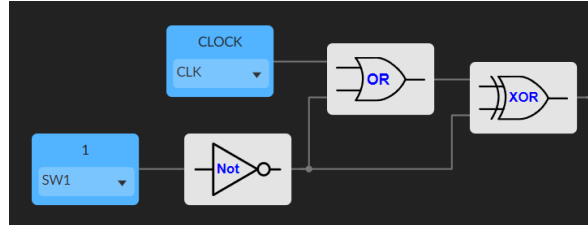
**Figur 6:** Tellerdesign i IceStudio.

I Figur 6 ser du hvordan telleren ble implementert i ice studio. Den ble implementert som en vanlig teller som inkrementerer med 1 hver *CLK* syklus. I tillegg har vi lagt til logikk så den skal begynne fra start igjen når den har kommet til 6.



**Figur 7:** LED-styring i IceStudio.

Led logikken ble også implementert i ice studio etter formlene funnet tidligere i Ligning 2, Ligning 3, Ligning 4, Ligning 5.



**Figur 8:** Klokkestyring visualisert

Og klokkestyringen ble implementert ifølge kravene spesifisert av Ligning 1.

## 3.2 Beregning av seriemotstander

For å beregne hvilke verdier vi trenger for de forskjellige motstanderne bruker vi formelen spesifisert i Ligning 6. Hver LED har forskjellig optimal spenning og strøm som skal føres over dioden, så tallverdiene for de forskjellige utregningene blir litt forskjellig. Hver resistor har en spesifisert motstand etter hvilken LED type de er koblet i serie med.

### 3.2.1 Rød LED

- $V_{LED}$  for rød LED: 2,0 V
- $I_{LED}$ : 20 mA

$$R_{\text{rød}} = \frac{3,3\text{ V} - 2,0\text{ V}}{0,020\text{ A}} = 65\ \Omega$$

### 3.2.2 Orange LED-er

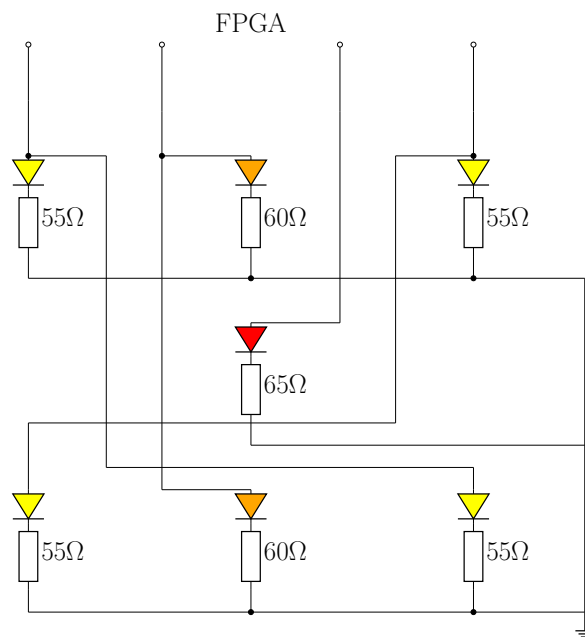
- $V_{LED}$  for orange LED: 2,1 V
- $I_{LED}$ : 20 mA

$$R_{\text{orange}} = \frac{3,3\text{ V} - 2,1\text{ V}}{0,020\text{ A}} = 60\ \Omega$$

### 3.2.3 Gule LED-er

- $V_{LED}$  for gul LED: 2,2 V
- $I_{LED}$ : 20 mA

$$R_{gul} = \frac{3,3 V - 2,2 V}{0,020 A} = 55 \Omega$$



**Figur 9:** Teorisert krets

I Figur 9 over ser du hvordan kretsen ønskelig skal bli koblet opp.

## 3.3 Total effektforbruk

Når vi nå vet mengden spenning og strøm over hver LED kan vi bruke dette til å beregne teoretisk effektforbruk. Igjen blir effekten til hver LED forskjellig etter hvilken type LED diode det er. Vi bruker Ligning 7 til å regne ut effektene for de forskjellige LED'ene.

### 3.3.1 Rød LED

$$P_{rød} = 2,0 V \cdot 0,020 A = 0,040 W$$

### 3.3.2 Orange LED-er

$$P_{\text{orange}} = 2,1 \text{ V} \cdot 0,020 \text{ A} = 0,042 \text{ W}$$

### 3.3.3 Gule LED-er

$$P_{\text{gul}} = 2,2 \text{ V} \cdot 0,020 \text{ A} = 0,044 \text{ W}$$

## 3.4 Effektforbruk for forskjellige terningkast

Effektforbruket avhenger av hvilke LED-er som er tent for hvert terningkast. Tabellen nedenfor viser effektforbruket for hvert utfall:

Terningkast	Tent LED-er	Total effektforbruk
1	Rød	0,04 W
2	2 gule	0,088 W
3	Rød + 2 gule	0,128 W
4	4 gule	0,176 W
5	Rød + 4 gule	0,216 W
6	Rød + 4 gule + 2 orange	0,3 W

## 3.5 Forventet effektforbruk

Forventet effektforbruk kan beregnes som gjennomsnittet av effektforbruket for alle seks utfall:

$$P_{\text{forventet}} = \frac{0,04 + 0,088 + 0,128 + 0,176 + 0,216 + 0,3}{6} = \frac{0,948}{6} \approx 0,158 \text{ W}$$

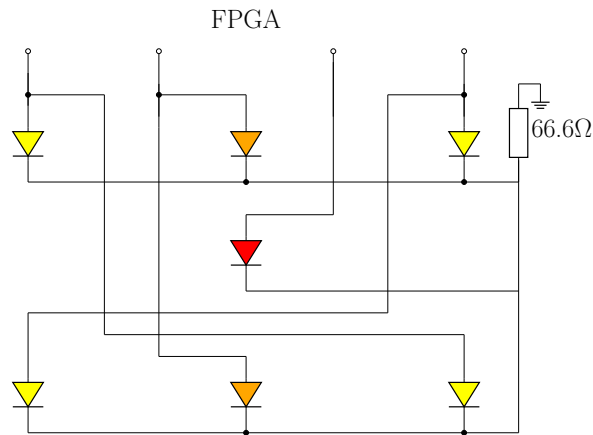
## 3.6 Implementering

I den opprinnelige designen ble hver LED koblet til en egen seriemotstand for å begrense strømmen. Imidlertid ble dette endret til en felles seriemotstand på  $66,6\Omega$  for å forenkle kretsen og redusere antall komponenter. Denne endringen ble gjort etter en grundig analyse av strømbegrensningene og LED-enes egenskaper. Selv om LED-ene får litt mindre energi enn det de er spesifisert for, lyser de fortsatt skarpt og tydelig, noe som viser at systemet fungerer godt selv med denne forenklede tilnærmingen.

## 3.7 Kretsdesign

Den nye kretsen består av følgende komponenter: - En Lattice ICE40 FPGA-brikke som styrer systemet. - Syv LED-er (fire gule, to orange, én rød) arrangert i et terningmønster og felles

seriemotstand for å begrense strømmen..



**Figur 10:** Kretskjema til det realiserte systemet

Resistorverdi på  $66.6\Omega$  ble valgt så den er litt høyere enn de teoretiske resistorverdiene. Dette forsikrer oss om at ingen av LED'ene brenner opp selv om all strømmen legger seg over en av dem.

### 3.8 Testprosedyre

Systemet ble testet ved å måle strømmen i kretsen for hvert mulig terningkast (1 til 6) og når knappen ble holdt nede. Målingene ble utført ved hjelp av et multimeter, og resultatene ble sammenlignet med de teoretiske beregningene for å verifisere at systemet oppførte seg som forventet. I tillegg ble LED-enes lysstyrke visuelt vurdert for å sikre at de lyser tilstrekkelig skarpt.

### 3.9 Testresultater

Tabellen nedenfor viser de målte strømmene for hvert terningkast, samt når knappen ble holdt nede:

Tilstand	Strøm (mA)
Knappen holdes nede	13,51
Terningkast 1	13,37
Terningkast 2	14,90
Terningkast 3	17,00
Terningkast 4	17,67
Terningkast 5	18,49
Terningkast 6	19,08

**Tabell 1:** Målte strømmen for hver tilstand.

### 3.10 Sammenligning av teoretisk og realisert effekt

Tabellen nedenfor sammenligner den teoretiske effekten (fra den opprinnelige tabellen) med den realiserte effekten (beregnet fra de målte strømmene). Effekten er beregnet ved å bruke Ligning 7 og den målte strømmen og spenningen.

Tilstand	Teoretisk effekt (W)	Realisert effekt (W)	Forskjell (W)
Knappen holdes nede	-	0,0446	-
Terningkast 1	0,04	0,0441	-0,0041
Terningkast 2	0,088	0,0492	0,0388
Terningkast 3	0,128	0,0561	0,0719
Terningkast 4	0,176	0,0583	0,1177
Terningkast 5	0,216	0,0610	0,1550
Terningkast 6	0,3	0,0630	0,2370

**Tabell 2:** Sammenligning av teoretisk og realisert effekt.

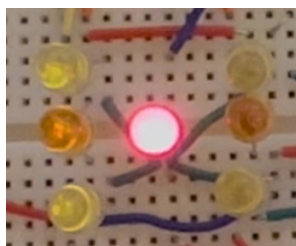
### 3.11 Diskusjon

Testresultatene viser at systemet fungerer som forventet, med et stabilt effektforbruk for hvert terningkast. Den felles seriemotstanden på  $66,6\Omega$  sikrer at strømmen gjennom hver LED er innenfor trygge grenser, og FPGA-en klarer å levere nok strøm til å drive alle LED-ene samtidig når det er nødvendig.

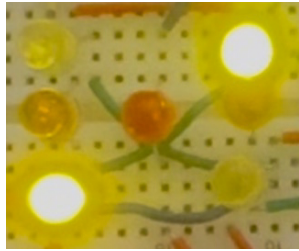
Vi ser en skarp økning i forskjell fra realisert til teoretisk effekt når flere LED lyser. Dette tilsier at når flere LED lyser samtidig, lyser hver individuelle led litt svakere enn det den ville gjort med kretsen teorisert i figur 2. Denne reduksjonen bidro helhetlig positivt ettersom den totale lysstyrken til terningen nå ikke lenger flukserer i like stor grad etter hvor mange LED som er tent.

En interessant observasjon er at LED-ene lyser skarpt og tydelig selv om de får litt mindre energi enn det de er spesifisert for. Dette skyldes sannsynligvis at LED-ene har en høy effektivitet og kan produsere tilstrekkelig lys selv ved lavere strøm. Dette gjør systemet mer energieffektivt uten å ofre funksjonalitet eller brukeropplevelse.

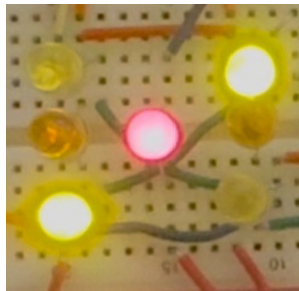
Under kan er bilder av terningen i alle de forskjellige verdiene.



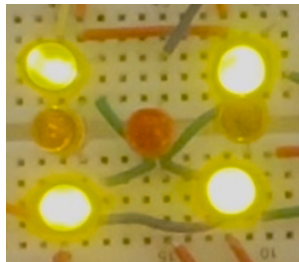
**Figur 11:** Terningkast 1



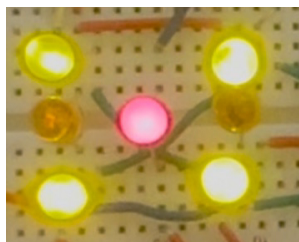
**Figur 12:** Terningkast 2



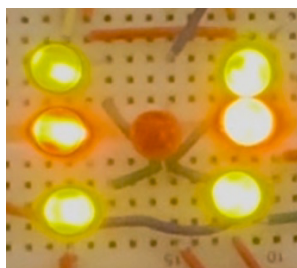
**Figur 13:** Terningkast 3



**Figur 14:** Terningkast 4

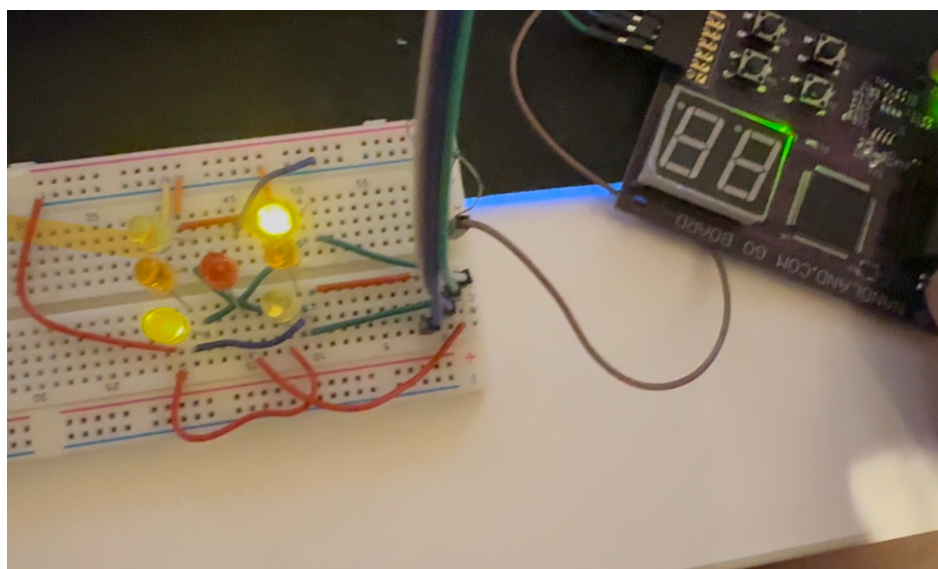


**Figur 15:** Terningkast 5



**Figur 16:** Terningkast 6

### 3.12 Komplet system



**Figur 17:** Fysisk realisering av systemet med felles seriemotstand.

Figur 17 viser den oppkoblede kretsen, den felles seriemotstanden gjør kretsen simpel og oversiktlig.

### 3.13 Fremtidige forbedringer

Selv om systemet fungerer godt, er det rom for forbedringer. For eksempel kan en mer nøyaktig strømbegrensning implementeres ved å bruke individuelle seriemotstander for hver LED. Dette vil sikre at hver LED får nøyaktig den strømmen den er spesifisert for, noe som kan forlenge levetiden til LED-ene og ytterligere forbedre systemets energieffektivitet. I tillegg kan en mer avansert tilfeldighetsteknikk implementeres for å gjøre terningkastene enda mer uforutsigbare.

## 4 Konklusjon

Designet av den elektroniske terningen basert på en Lattice ICE40 FPGA har vist seg å være vellykket. Systemet oppfyller alle de funksjonelle og ikke-funksjonelle kravene, inkludert tilfeldig generering av terningkast, korrekt lysmønster for hvert utfall, og et stabilt effektforbruk. Testresultatene bekrefter at effektforbruket ligger innenfor de teoretiske beregningene. Den nye kretsen med felles seriemotstand har vist seg å være en effektiv og enkel løsning som reduserer kompleksiteten uten å ofre ytelse.

Systemet er enkelt å bruke, med en trykknapp som aktiverer terningkastet, og LED-ene lyser tydelig og korrekt for hvert utfall. Selv om LED-ene får litt mindre strøm enn spesifisert, er lysstyrken tilstrekkelig for formålet.