



# **SISTEM DE ILUMINARE VARIABILĂ**

**FPGA Basys3 – XC7A35T-ICPG236C**

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

**ANUL I DE STUDIU, GRUPA 30218**

proiect realizat de Borșa Iulian-Laurențiu

**Profesor îndrumător:**

Pop Diana



---

**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

---



|  | pagina   |
|--|----------|
| <b>CUPRINS.....</b>  | <b>3</b> |
| 1. Specificația proiectului.....   | 4        |
| 2. Schema bloc cu componentele principale.....   | 7        |
| 3. Evidențierea unității de comandă și a celei de execuție.....                          | 8        |
| 4. Etapele de proiectare:  |          |
| a. Descrierea formală a principalelor componente.....                                    | 8        |
| b. Codificarea stărilor.....   | 9        |
| c. Minimizarea ecuațiilor logice aferente.....   | 9        |
| d. Schemele logice pentru elementele componente și pentru ansamblul proiectului.....     | 11       |
| 5. Lista componentelor utilizate.....  | 11       |
| 6. Semnificația notațiilor efectuate în proiect și a interfeței cu exteriorul (I/O)..... | 14       |
| 7. Justificarea soluției alese.....  | 15       |
| 8. Instrucțiuni de utilizare și întreținere.....   | 15       |
| 9. Posibilități de dezvoltare ulterioară.....  | 16       |
| 10. Referințe.....   | 17       |



## 1. Specificația proiectului

Să se proiecteze un **sistem de iluminare variabilă** cu ledurile de pe plăcile cu FPGA. Sistemul va avea mai multe moduri de funcționare:

- Mod manual – valoarea intensității luminoase a ledurilor se furnizează de pe întrerupătoare (8 bit);
- Mod test – intensitatea luminoasă a ledurilor variază de la valoarea minimă la valoarea maximă într-un interval de timp specific fiecărui led (led0 - 1 secundă, led1 - 2 secunde, ..., led7 - 8 secunde), forma de undă aproximată fierăstrău;
- Mod automat – intensitatea luminoasă a ledurilor variază de la valoarea minimă la valoarea maximă și înapoi la minim într-un interval de timp măsurat în secunde. Acest interval de timp este furnizat ca o intrare a sistemului, forma de undă aproximată triunghi.

Pentru varierea intensității luminoase se va folosi tehnica PWM (pulse width modulation). Documentație: manualul de referință al plăcii cu FPGA ([www.digilentinc.com](http://www.digilentinc.com)).

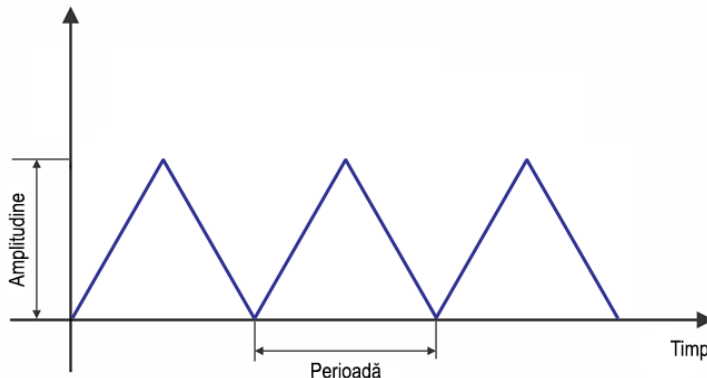




## 1.1. Forme de undă

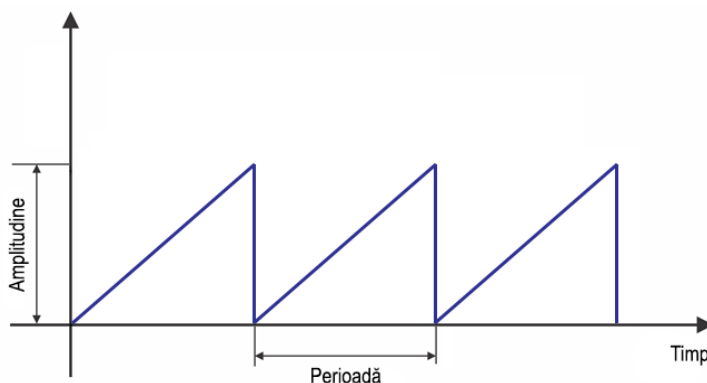
### a) Forma de undă triunghi

Intensitatea luminoasă a ledurilor variază de la valoarea minimă la valoarea maximă și înapoi la valoarea minimă într-un interval de timp numit perioadă. Atunci când timpul este egal cu jumătate din perioadă, intensitatea luminoasă a ledurilor este maximă, iar când timpul este egal cu extremele perioadei, intensitatea luminoasă a ledurilor este minimă.



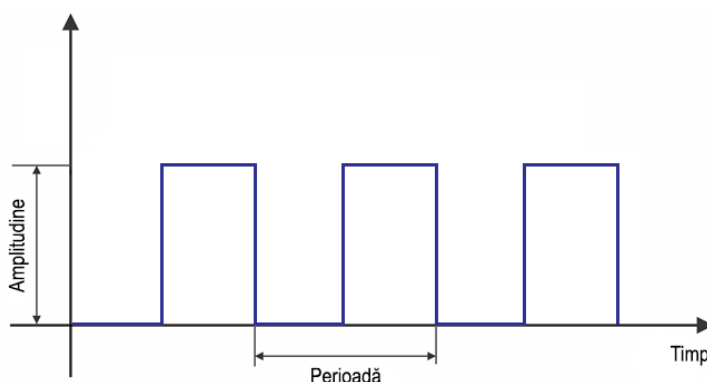
### b) Forma de undă fierăstrău

Intensitatea luminoasă a ledurilor variază de la valoarea minimă la valoarea maximă într-un interval de timp numit perioadă. La o extremă a perioadei intensitatea luminoasă a ledurilor este minimă, iar la cealaltă extremă a perioadei intensitatea luminoasă a ledurilor este maximă. După ce se ajunge la intensitatea maximă, următoarea stare este intensitatea minimă.



### c) Forma de undă pătrat

Forma de undă pătrat este o formă de undă periodică non-sinusoidală, în care amplitudinea alternează pentru o frecvență constantă între o valoare minimă și una maximă fixă, cu aceeași durată la minim și maxim. Trecerea de la minim la maxim este instantanee pentru forma de undă pătrat ideală. Acest lucru nu este realizabil în sistemele fizice. Formele de undă pătate sunt adesea întâlnite în domeniul procesării electronice și a semnalelor. O formă de undă simetrică, cu durată arbitrară la minim și maxim, se numește **pulse wave**.



Raportul perioadei maxime cu perioada totală a oricărei forme de undă dreptunghiulară se numește **duty cycle**. O formă de undă ideală are un ciclu de funcționare de 50% (perioade egale pentru minim și maxim). Nivelul mediu al formei de undă dreptunghiulară este, de asemenea, dat de ciclul de funcționare, prin urmare prin modificarea perioadelor de pornire și oprire și apoi prin medierea acestor perioade, este posibilă reprezentarea oricărei valori între cele două niveluri limită. Aceasta stă la baza fenomenului **PWM (Pulse Width Modulation)**.



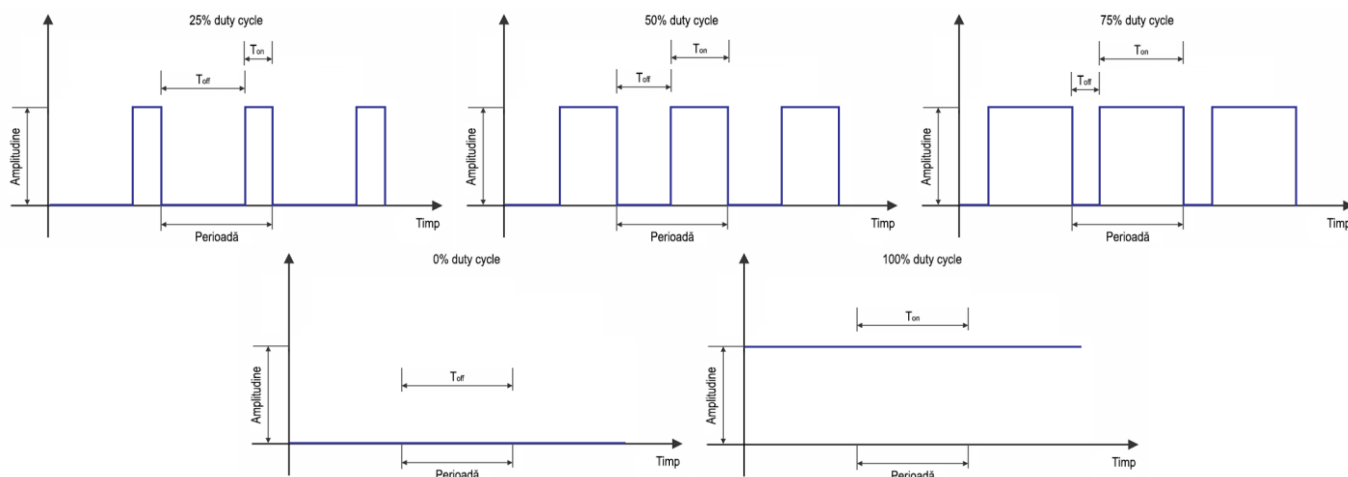
## 1.2. Tehnica PWM (Pulse Width Modulation)

PWM (**Puls Width Modulation**) este o tehnică folosită pentru a varia în mod controlat tensiunea dată unui dispozitiv electronic. Această metodă schimbă foarte rapid tensiunea oferită dispozitivului respectiv din ON în OFF și invers (tregeri rapide din HIGH (5V de exemplu) în LOW (0V)). Perioada de timp corespunzătoare valorii ON dintr-un ciclu ON-OFF se numește factor de umplere (**duty cycle**) și reprezintă, în medie, ce tensiune va primi dispozitivul electronic. Astfel, se pot controla circuite analogice din domeniul digital. Practic, asta înseamnă că un LED acționat astfel se va putea aprinde / stinge gradual, iar în cazul unui motor acesta se va învârti mai repede sau mai încet.

### Exemple:

✓ În cazul unui LED, căruia i se aplică un semnal PWM cu factor de umplere de 0%, intensitatea luminoasă va fi 0, iar LED-ul va fi stâns. Un factor de umplere de 100% va duce la o intensitate luminoasă maximă a acestuia.

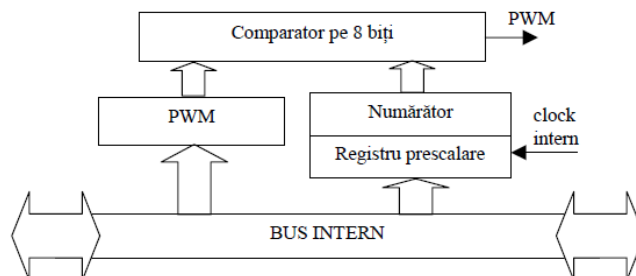
✓ În cazul unui motor, căruia i se aplică un semnal PWM cu factor de umplere de 0%, viteza de rotație a acestuia va fi egală cu 0rpm. Un factor de umplere de 100% va duce la o turație maximă a acestuia.



Semnalul PWM este un semnal periodic la care se poate modifica în mod controlat factorul de umplere.

Structura unui canal PWM este reprezentată în figură.

Registru de prescalare generează clock-ul pentru un numărator. Clock-ul de numărare este programabil, iar conținutul număratorului este comparat cu cel al componentei PWM. Cât timp rezultatul comparației este mai mic sau egal, se va genera un '1' logic la ieșirea PWM. Dacă rezultatul comparației este mai mare, se va genera la ieșirea PWM un '0' logic.



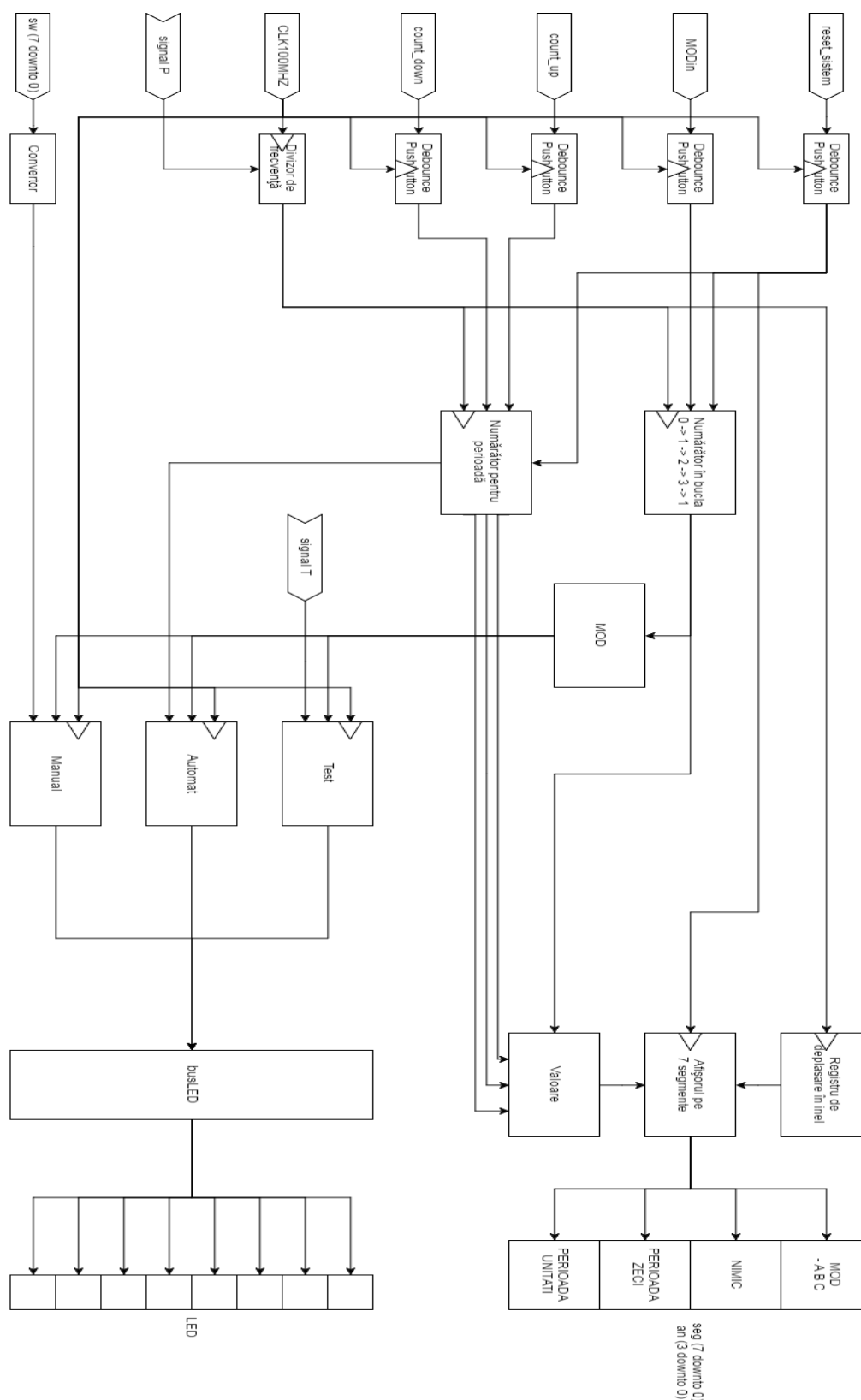
Din figure se observă că  $T_{on}$  reprezintă timpul în care ledurile sunt aprinse, iar  $T_{off}$  timpul în care sunt stânse.

$$T_{total} = T_{on} + T_{off}$$

$$D = \frac{T_{on}}{(T_{on} + T_{off})} = \frac{T_{on}}{T_{total}}$$



## 2. Schema bloc cu componentele principale





### 3. Evidențierea unității de comandă și a celei de execuție

Schema bloc a sistemului de iluminare variabilă este compusă din două unități, mai exact din: Unitatea de Comandă (UC) și Unitatea de Execuție (UE).

Unitatea de comandă și control are rolul de a genera succesiunea semnalelor de comandă care asigură secvența corectă de execuție a operațiilor. Această succesiune este specifică fiecărei instrucțiuni, fiind determinată atât de codul operației, cât și de recepționarea unor semnale de stare de la circuitele controlate, prin care se verifică îndeplinirea unor condiții.

În unitatea de comandă sunt regăsite componentele:

- Numărător în bucla 01231
- Registru de deplasare în inel
- Convertor din binar în integer
- Numărător pentru perioadă
- Multiplexor
- Decodificator

Toate comenzile, indiferent de natura lor sunt analizate în unitatea de comandă și mai apoi transmise către unitatea de execuție. În funcție de comenzi, unitatea de execuție va genera semnale pentru următoarele componente:

- Afișorul pe 7 segmente
- Componenta automata
- Componenta manuală
- Componenta test

Mai departe, unitatea de execuție va decide care mod va fi folosit, ce valori să apară pe afișor și care leduri să pulseze. Totodată se stabilește și forma de undă aproximată.

### 4. Etapele de proiectare

#### 4.a. Descrierea formală a principalelor componente

Componenta manuală este formată dintr-un numărător modulo 256, care va număra de la 0 la 255. Acestu numărător are limita superioară 255 deoarece folosim 8 biți de intrare pentru componenta manuală și cel mai mare număr care poate fi scris în binar este “11111111”, adică 255. În interiorul componentei mai există și un comparator, care va avea mereu ca intrări ieșirea de la numărător și numărul convertit ca și întreg, introdus de la întrerupătoare.

Componenta automată este formată dintr-un numărător reversibil care numără de la 0 la 255 într-o jumătate de perioadă, iar de la 255 înapoi la 0 în a doua jumătate de perioadă. În rest, totul funcționează ca și la componenta manuală. Componenta automată mai are un numărător care trebuie să se asigure că în 256 de stări, intensitatea luminoasă trebuie să crească de la 0% la 100%.

Din acest motiv și având în vedere că pe plăcile cu FPGA Basys3, clock-ul implicit are o frecvență de 100MHz, am dedus că sistemul operează la 10ns.





Pentru a afla limita superioară de numărare la care trebuie să se ajungă într-o secundă, perioada oscilatorului trebuie împărțită la 256 (astfel se atinge intensitatea luminoasă a ledurilor de 100%). La final se mai împarte încă o dată la 2, pentru a avea jumătate de perioadă. Rezultatul va fi 195312.

Componenta test va funcționa asemănător cu celelalte două componente, deoarece este o combinație între acestea. Singura diferență dintre ele este că la component test se folosește un numărător direct. Aceleași calcule vor fi efectuate și în cazul componente test, doară că la aceasta limita de numărare nu se va mai împărți la 2, ca și în cazul anterior, ci va rămâne așa, deoarece forma de undă aproximată este fierăstrău, nu triunghi.

#### 4.b. Codificarea stărilor

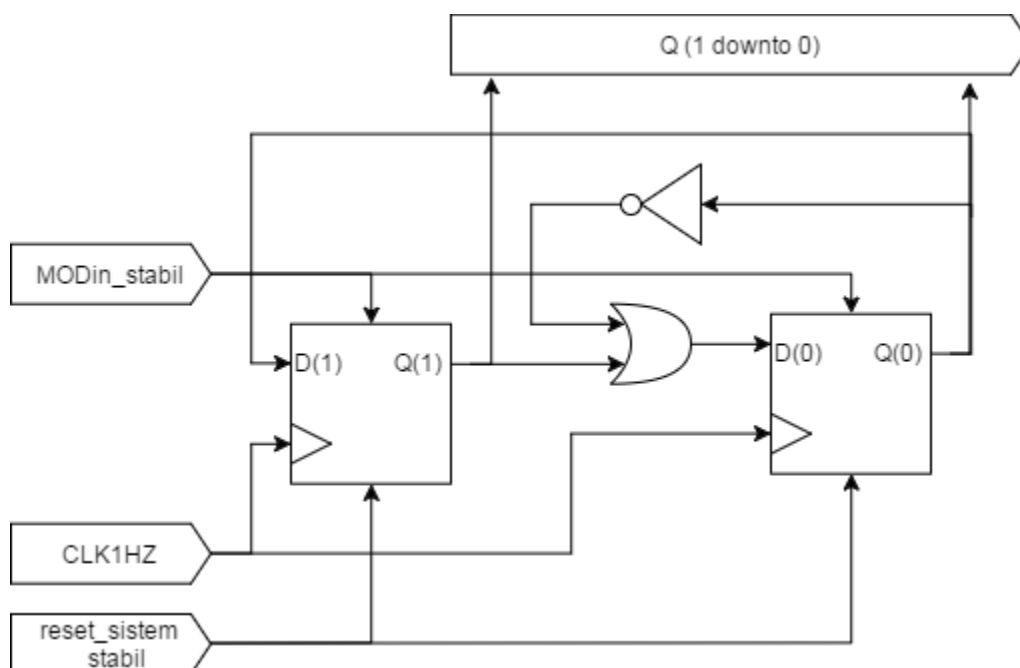
Având în vedere că sistemul de iluminare variabilă funcționează după 3 moduri, am decis să folosesc un cuvânt de date pe 2 biți, fiecare dintre cele 4 posibilități fiind un enable pentru unitatea de execuție.

- “00” – stare RESET
- “01” – mod MANUAL
- “10” – mod TEST
- “11” – mod AUTOMAT

#### 4.c. Minimizarea ecuațiilor logice aferente

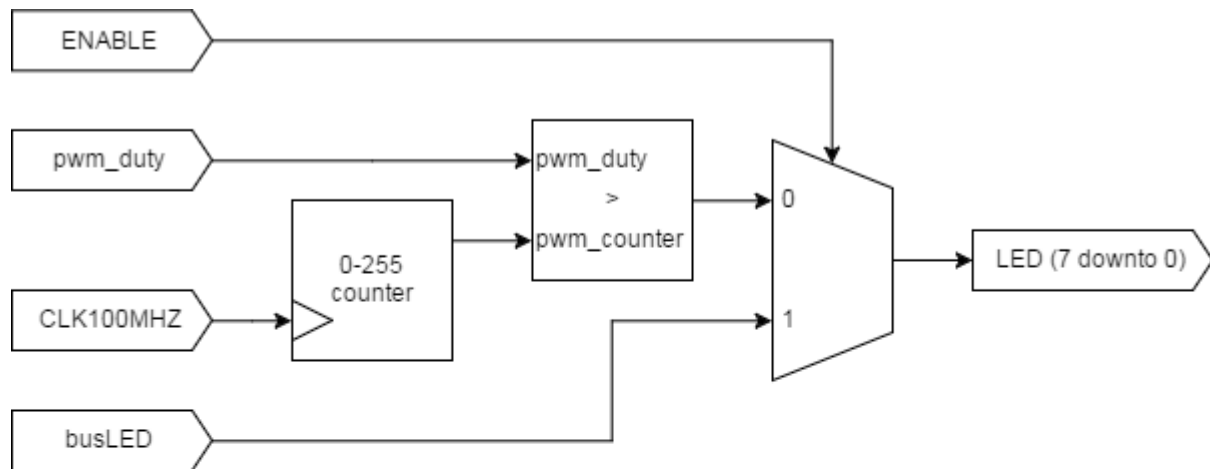
#### 4.d. Scheme logice pentru elementele componente și pentru ansamblul proiectului

- Schema logică a numărătorului în bucla 0->1->2->3->1

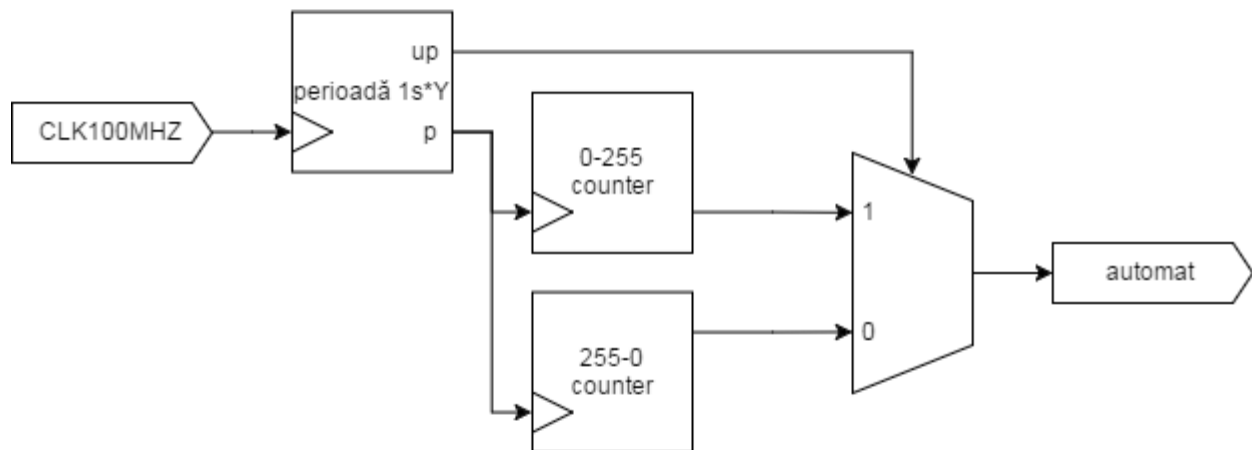




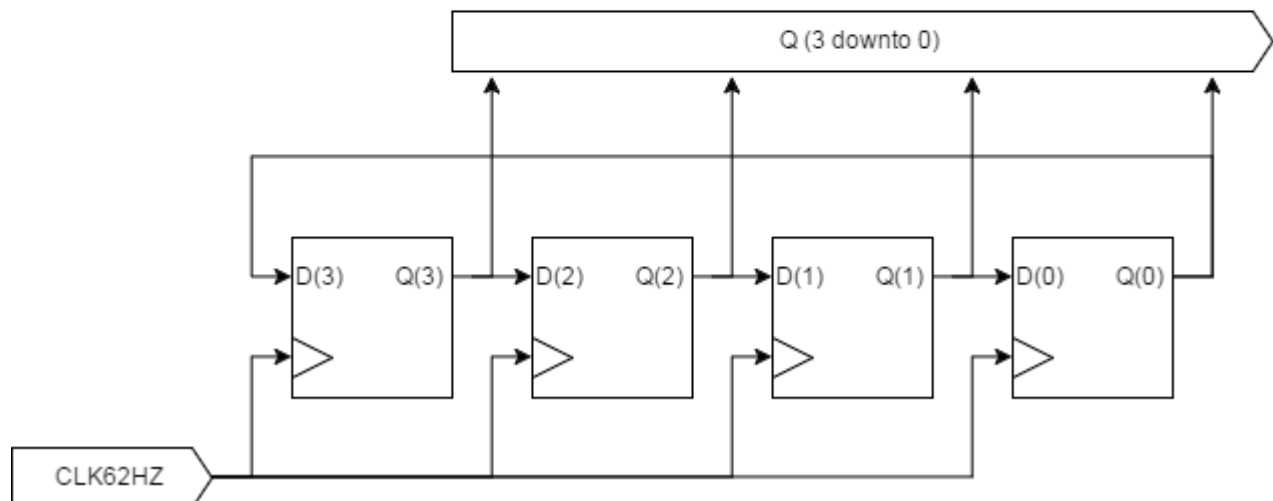
- Schema logică a componentei manuale



- Schema logică a componentei automate



- Schema logică a componentei automate





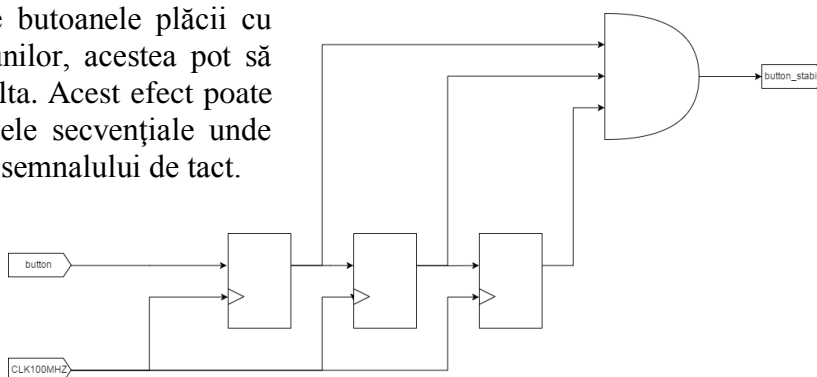
## 5. Lista componentelor utilizate

### 5.1. Debounce Pushbutton

Atunci când apeși unul dintre butoanele plăcii cu FPGA Basys3, din cauza imperfecțiunilor, acestea pot să facă sistemul să sară dintr-o stare în alta. Acest efect poate să fie o problemă serioasă în circuitele secvențiale unde acțiunea are loc pe frontul crescător al semnalului de tact.

Din acest motiv este necesar un circuit pentru debounce când folosim circuite secvențiale.

Rolul circuitului este de a stabili semnalul transmis de un buton al plăcii cu FPGA Basys3.

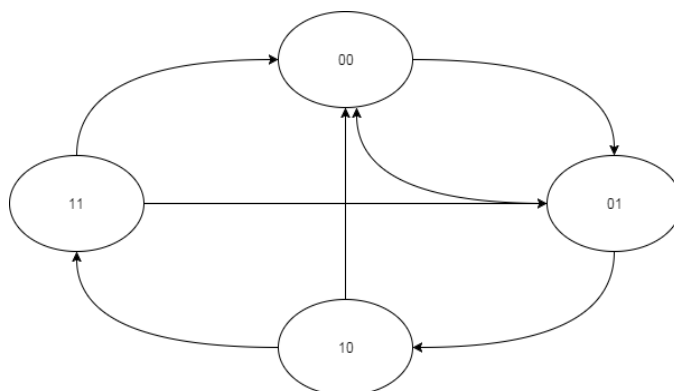


### 5.2. Numărător în bucla 0->1->2->3->1

Această componentă este folosită pentru a selecta modul de funcționare a sistemului de iluminare variabilă. Dat fiind faptul că sistemul are trei moduri, manual, automat și test, va fi nevoie de trei stări diferite plus o stare în care va ajunge circuitul după resetarea asincronă.

Fiecărei stări îi corespunde un mod:

- ✓ 00 – Mod RESET
- ✓ 01 – Mod MANUAL
- ✓ 10 – Mod TEST
- ✓ 11 – Mod AUTOMAT



### 5.3. Numărător pentru perioadă

Pentru modul automat este nevoie de un numărator cu ajutorul căruia să decidem în ce perioadă de timp, componenta automată va realiza ciclul de la valoarea minimă a intensității luminoase până la valoarea maximă a intensității luminoase și înapoi, după forma de undă aproximată triunghi, detaliată în primul capitol, la formele de undă. Intervalul de timp va fi selectat de la butoanele plăcii cu FPGA Basys 3, acesta putând fi crescut sau scăzut în intervalul 0-99. Timpul va fi măsurat în secunde.

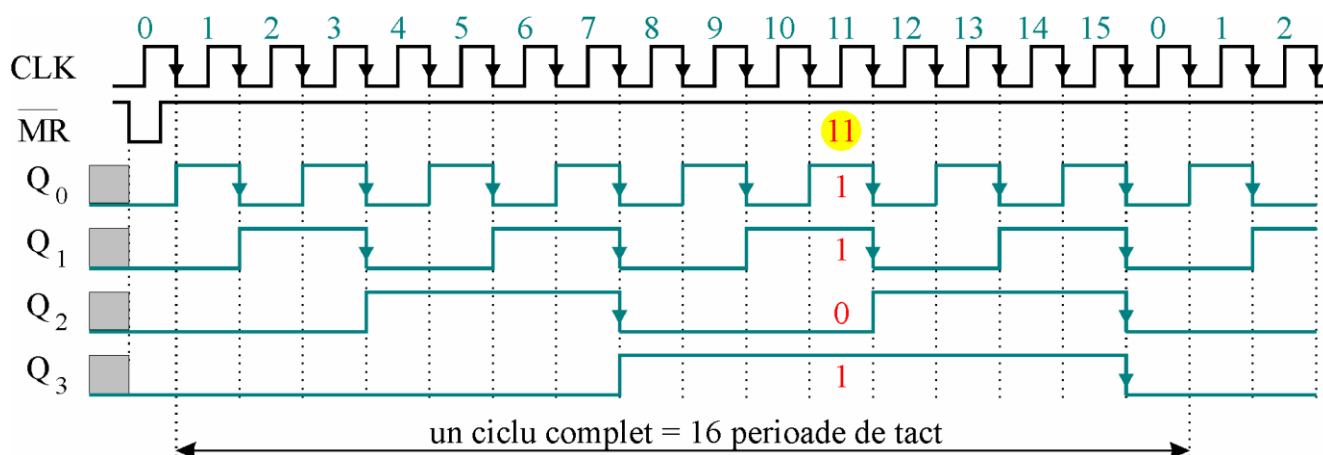
### 5.4. Divizor de frecvență

Orice numărator este în același timp un divizor de frecvență, raportul de divizare fiind chiar modulul p al număratorului. Cel mai simplu caz este cel cu divizare cu un raport fix, iar cel mai flexi-



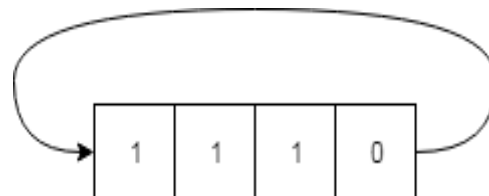
bil – divizorul programabil de frecvență, la care raportul de divizare este specificat din exterior și se poate modifica oricând. Este important de arătat că nu întotdeauna factorul de umplere la ieșirea divizorului este  $\frac{1}{2}$ , existând aplicații în care acest aspect nu este deranjant.

Pentru acest proiect am folosit un divizor de frecvență programabil deoarece pentru unele componente voi avea nevoie de o frecvență de 1Hz, iar pentru altele 62,5Hz.



### 5.5. Registru de deplasare în inel

Acest circuit ajută la buna funcționare a afișorului pe 7 segmente, mai exact, cu ajutorul lui reușim să “păcălim” ochiul și să transmitem anodului un semnal cu frecvența de 62,5Hz. Regimul de funcționare a registrului este shiftarea în inel a unui STD\_LOGIC\_VECTOR:=”1110”.



### 5.6. Afișorul pe 7 segmente

Afișorul pe 7 segmente este un ansamblu format din 4 afișoare cu câte 7 leduri conectate la 4 anodi printr-un catod. Pentru ca fiecare afișor să îi apară utilizatorului ca fiind corespunzător (ca luminozitate) și continuu iluminat fiecare din cele 4 cifre (digits), trebuie să fie activă (aprinsă) la un interval de timp (refresh period) cuprins între 1ms și 16ms.

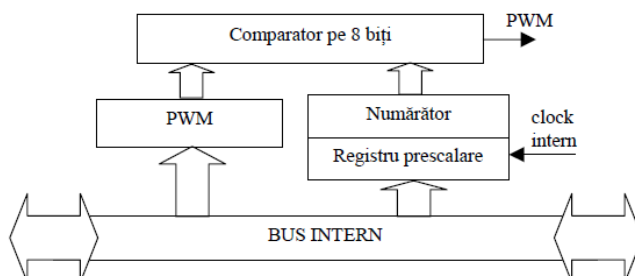
Cu alte cuvinte frecvența de înprospătare (refresh frequency) a informației pentru fiecare cifră trebuie să fie între 1KHz și 62.5Hz. De exemplu, dacă frecvența de înprospătare este de 62.5Hz (16ms) fiecare cifră va fi aprinsă (într-un ciclu de 16ms, pentru 4 afișoare) pentru un interval de 4 ms (digit period). Utilizatorul trebuie să asigure periodizarea și secvențierea corectă a comenzilor pentru anodi și segmente (corelația temporală corectă de tip AN<sub>x</sub> – Digit<sub>y</sub>).

Singurul oscilator de ceas disponibil pe placa cu FPGA Basys3 este unul de 100MHz a cărui frecvență este mult prea mare pentru a genera periodizarea necesară în sensul celor descrise anterior. Astfel vom utiliza un numărător care va funcționa pe post de divizor de frecvență. Ieșirile pentru anodi vor fi generate de registru de deplasare în inel, prezentat mai sus.



### 5.7. Componenta manuală

Componenta manuală este una dintre componentele fără de care sistemul de iluminare variabil nu ar funcționa. Aceasta este formată dintr-un numărator integrat, care numără de la 0 la 255. Limita de numărare este 255 deoarece intrările acestei componente sunt pe 8 biți. Tot în interiorul componentei manuale se mai găsește un comparator, care va decide când vor fi aprinse ledurile și când nu. În acest mod se calculează factorul de umplere și ledurile vor avea o intensitate luminoasă care variază în funcție de intrările (switch-uri) componente.



### 5.8. Componenta automată

Componenta automată funcționează identic cu componenta manuală, doar că în interiorul acesteia mai există un numărator reversibil care are rolul de a varia intensitatea luminoasă a ledurilor între valoarea minimă și maximă. Numărătorul are rolul de a schimba factorul de umplere de la minim la maxim într-o perioadă de timp. Componenta se numește automată deoarece factorul de umplere este modificat de un numărator, ca și cum ar exista un om care să schimbe valorile de pe intrările componente automate într-o manieră ideală.

Această componentă va face sistemul să funcționeze după forma de undă aproximată triunghi.

### 5.9. Componenta test

Componenta test este o combinație între celelalte două componente. Modul său de funcționare este același pentru fiecare dintre cele 8 leduri, doar că perioada fiecăruia va fi egală cu numărul său de ordine. (Clarificare: pentru primul led, perioada este de o secundă, iar pentru ultimul led, perioada este de 8 secunde)

Această componentă va face sistemul să funcționeze după forma de undă aproximată fierăstrău.

### 5.8. Alte componente utilizate

- Porți logice
- Multiplexoare
- Decodificatoare
- Convertoare între **INTEGER** și **STD\_LOGIC\_VECTOR**
- Bistabile



## 6. Semnificația notațiilor efectuate în proiect și a interfeței cu exteriorul (I/O)

| NUME         | TIP             | NET                                    | ROL  |
|--------------|-----------------|--|--|
| reset_sistem | intrare (1 bit) | U17                                    | resetează sistemul   |
| MODin        | intrare (1 bit) | U18                                    | schimbare mod  |
| count_up     | intrare (1 bit) | T17                                    | crște perioada la modul AUTOMAT  |
| count_down   | intrare (1 bit) | W19                                    | reduce perioada la modul AUTOMAT                                       |
| CLK100MHZ    | intrare (1 bit) | W5                                     | clock de la placă  |
| sw           | intrare (8 bit) | W13, W14, V15, W15, W17, W16, V16, V17 | selectarea factorului de umplere pentru modul MANUAL                   |
| LED          | ieșire (8 bit)  | V14, U14, U15, W18, V19, U19, E19, U16 | afișarea intensității luminoase pe leduri                              |
| seg          | ieșire (7 bit)  | U7, V5, U5, V8, U8, W6, W7             | afișarea cifrelor și literelor de la modurile folosite pe fiecare anod |
| an           | ieșire (4 bit)  | W4, V4, U4, U2                         | funcționarea afișorului  |

De asemenea, proiectul are un număr mare de semnale, prin care s-a făcut legătura între componenta principală și celelalte componente. La rândul ei, fiecare componentă conține semnale care ajută la scrierea corectă din punct de vedere sintactic al codului.

Un exemplu de semnale ar putea fi chiar semnalele folosite în componenta principală:

```

signal Y: INTEGER;
signal manual, busLED: STD_LOGIC_VECTOR (7 downto 0);
signal int_manual, int_test, int_automat: INTEGER;
signal pwm_counter, pwm_duty: INTEGER :=0;
signal MODout: STD_LOGIC_VECTOR (1 downto 0);
signal MODin_stabil, count_up_stabil, count_down_stabil, reset_sistem_stabil: STD_LOGIC;
signal u,z,s,m : STD_LOGIC_VECTOR(3 downto 0);
signal value : STD_LOGIC_VECTOR(15 downto 0);
signal test1,test2,test3,test4,test5,test6,test7,test8 : INTEGER :=0;
signal CLK1HZ, CLK62HZ : STD_LOGIC;
signal Qregistru : STD_LOGIC_VECTOR (3 downto 0);
signal LED0,LED1,LED2,LED3,LED4,LED5,LED6,LED7: STD_LOGIC;
signal enable: STD_LOGIC;

```

Se poate observa că unele intrări din entitate se regăsesc în lista de semnale, cu același nume, doar că la final este adăugată terminația „\_stabil”. Rolul lor este de a stabili semnalul transmis de la buton, fiind ieșiri pentru componenta Debounce Pushbuttons.

Semnalele CLK sunt folosite pentru a transmite mai departe clock-ul de la placă divizat, pentru componente cum ar fi Numărătoarele sau Afișorul pe 7 segmente.

Semnalele LEDX și testX sunt transmise la componenta care realizează modul TEST.

Celelalte semnale intră în funcționarea celorlalte componente precum registrul de deplasare în inel sau a afișorului.





## 7. Justificarea soluției alese

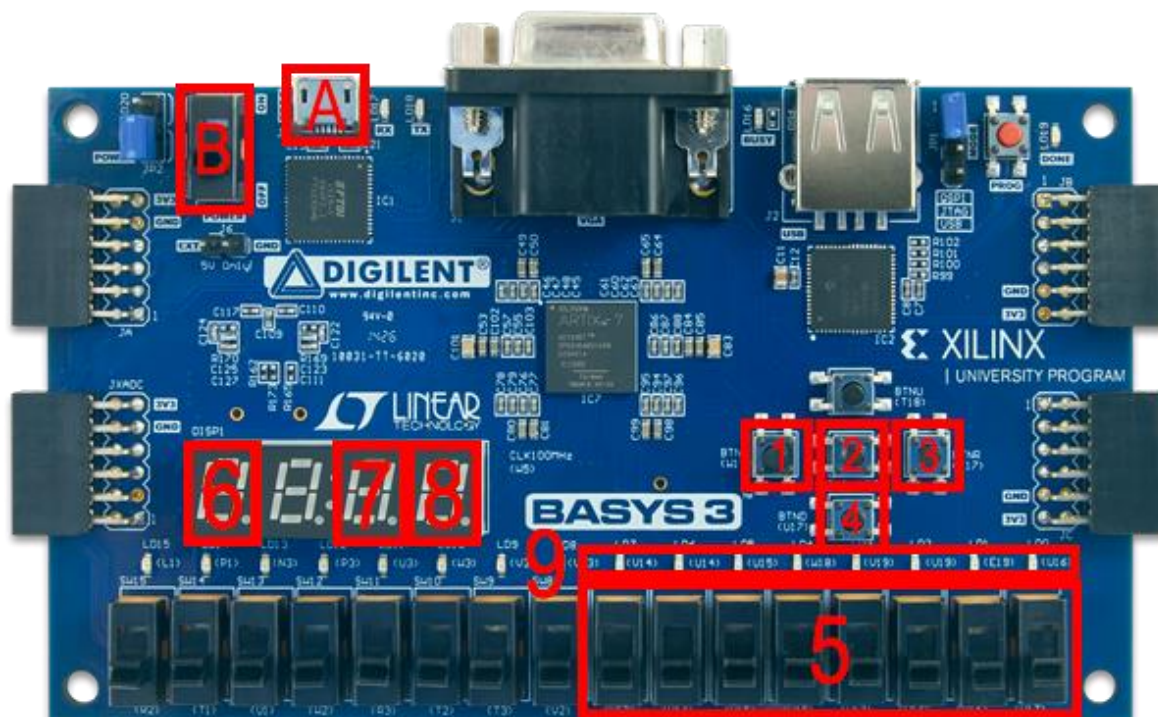
Nu am știut absolut nimic despre acest proiect atunci când l-am ales. Mi-ar fi plăcut să iau automatul de bilete pentru tren, dar a fost ales deja de o altă colegă. Prin urmare, am decis să iau un proiect oarecare.

În primele săptămâni am citit zeci de pagini de documentații despre PWM și modul său de funcționare până să aflu cum funcționează el de fapt. Acum că am reușit să termin proiectul, pot să spun că nu am făcut o greșală când a ales să realizez un proiect despre care nu știam nimic, deoarece între timp, citind documentații, am dat peste o mulțime de informații care mi-ar putea fi utile în continuare.

## 8. Instrucțiuni de utilizare și întreținere

Primii pași care trebuie îndepliniți pentru a putea folosi placa cu FPGA Basys3 sunt următorii:

- A. Se face conexiunea între placă și calculator printr-un Micro USB
- B. De la întrerupător, se trece din starea OFF în starea ON
- C. Din VIVADO se încarcă bitstream-ul pe placă





În cele ce urmează, voi prezenta modul de funcționare al sistemului:

- ✓ De pe butonul **2** se va selecta modul de funcționare. Pentru a trece într-o altă stare, se va acționa butonul **2**, care va duce sistemul în modul următor sau butonul **4**, care va reseta sistemul.
- ✓ După modul AUTOMAT, la următorul tact pe butonul **2** va urma modul MANUAL
- ✓ În toate cele 3 moduri, intensitatea luminoasă va varia pe ledurile **9**

A. după primul tact de la încărcarea proiectului pe placă, sistemul va trece din starea de RESET în modul MANUAL

- pe afișorul cu 7 segmente, cel mai din stânga anod **6** va afișa modul în care se găsește sistemul, adică **A**
- de la întrerupătoarele **5** se stabilește intensitatea luminoasă a ledurilor

B. după al doilea tact de la încărcarea proiectului pe placă, sistemul va trece din modul MANUAL în modul TEST

- pe afișorul cu 7 segmente, cel mai din stânga anod **6** va afișa modul în care se găsește sistemul, adică **B**

C. după al treilea tact de la încărcarea proiectului pe placă, sistemul va trece din modul TEST în modul AUTOMAT

- pe afișorul cu 7 segmente, cel mai din stânga anod **6** va afișa modul în care se găsește sistemul, adică **C**
- pe afișorul cu 7 segmente, cel mai din dreapta anod **8** va afișa unitățile, iar anodul din stânga lui **7** va afișa zecile pentru perioada de timp introdusă de pe butoanele **1** și **3**
- de la butonul **3** vom crește perioada de timp în care să pulseze ledurile, iar de la butonul **1** vom reduce perioada

## 9. Posibilități de dezvoltare ulterioară

- ✓ Pentru modul MANUAL, pe afișorul cu 7 segmente poate fi afișat factorul de umplere selectat de pe întrerupătoare
- ✓ Pentru dezvoltare, la placa cu FPGA Basys3 se poate conecta o tastatură de pe care să se introducă factorul de umplere al ledurilor. Asemenea funcției Fn+F3 de pe tastatura unui laptop, se poate alege o combinație de taste de pe care să se crească sau să se reducă intensitatea luminoasă a ledurilor.
- ✓ Tot de la tastatură ar putea fi schimbat și modul de funcționare





## 10. Referințe