**UNIVERSITATEA ”TRANSILVANIA„ DIN BRAȘOV**

*Facultatea de Inginerie Electrică și Știința Calculatoarelor*

*Specializarea: Tehnologii și Sisteme de Telecomunicații*

**PROIECT**

***Multiplicator cu deplasare la stânga***

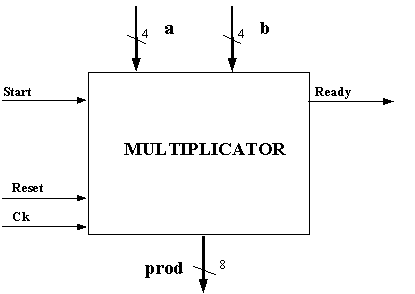
**Profesor coordonator: Student:**

***Sef lucr. dr. Traian TULBURE Deordica Bianca***

**-2013/2014-**

**Specificatii de nivel inalt**

* Schema bloc a multiplicatorului:



**Specificații de proiectare**

* Descrierea porturilor:

|  |  |  |  |
| --- | --- | --- | --- |
| **Denumire port** | **Sens** | **Dim[biti]** | **Semnificatie** |
| op1 | in | 4 | Intrare pe 4 biti reprezentand primul operand |
| op2 | in | 4 | Intrare pe 4 biti reprezentand cel de-al doilea operand |
| rezultat | out | 8 | Iesire pe 8 biti reprezentand rezultatul(produs pe 8 biti, la inmultire) |
| Start | in | 1 | Semnal pentru startul operatiei de inmultire |
| valid | out | 1 | Iesire care indica existenta rezultatului valid al operatiei de inmultire |
| reset\_n | in | 1 | Semnal de reset asincron, activ in 0 |
| Ck | in | 1 | Semnal de ceas |

* Se va utiliza urmatorul algoritm:

*Algoritm de inmultire cu deplasare stinga*

- reseteaza P (2\*n biti)

- incarca deinmultitul in A (n biti)

- incarca inmultitorul in B (n biti)

- repeta de n ori

- deplaseaza cu o pozitie stinga P (LSB P = 0)

- daca MSB A = 1 atunci

- P <= P + B

- deplaseaza cu o pozitie stinga A (LSB A = X)

- P contine produs

*Exemplu:*

233 x 195 = 45435

233 = 1110 1001

195 = 1100 0011

Reg. P Reg. A

----------- ---------

0000 0000 0000 0000 *1*110 1001 - initial

0000 0000 0000 0000

1100 0011

-------------------

0000 0000 1100 0011

0000 0001 1000 0110 *1*101 001X - iteratia 1

0000 0001 1000 0110

1100 0011

-------------------

0000 0010 0100 1001

0000 0100 1001 0010 *1*010 01XX - iteratia 2

0000 0100 1001 0010

1100 0011

-------------------

0000 0101 0101 0101

0000 1010 1010 1010 *0*100 1XXX - iteratia 3

0001 0101 0101 0100 *1*001 XXXX - iteratia 4

0001 0101 0101 0100

1100 0011

-------------------

0001 0110 0001 0111

0010 1100 0010 1110 *0*01X XXXX - iteratia 5

0101 1000 0101 1100 *0*1XX XXXX - iteratia 6

1011 0000 1011 1000 *1*XXX XXXX - iteratia 7

1011 0000 1011 1000

1100 0011

-------------------

**1011 0001 0111 1011**

**-------------------**

**Produs = 11x4096 + 1x256 + 7x16 + 11 = 45435**

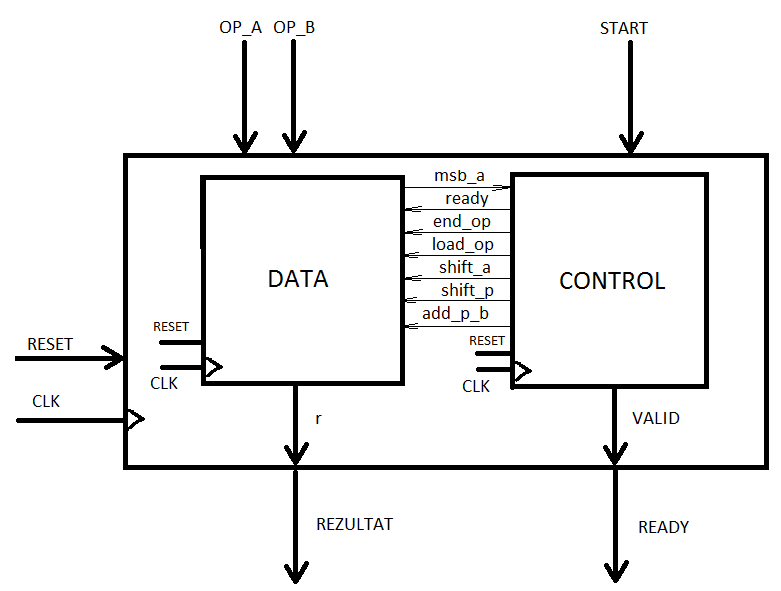
## Constrangeri impuse

Sistemul se va modela ca un sistem riguros sincron. Semnalul ***ck*** va fi primit de catre toate registrele din sistem. Se vor evita circuitele de divizare a frecventei semnalului de tact. Se vor folosi, eventual, circuite de generare a semnalelor de activare (*enable*).

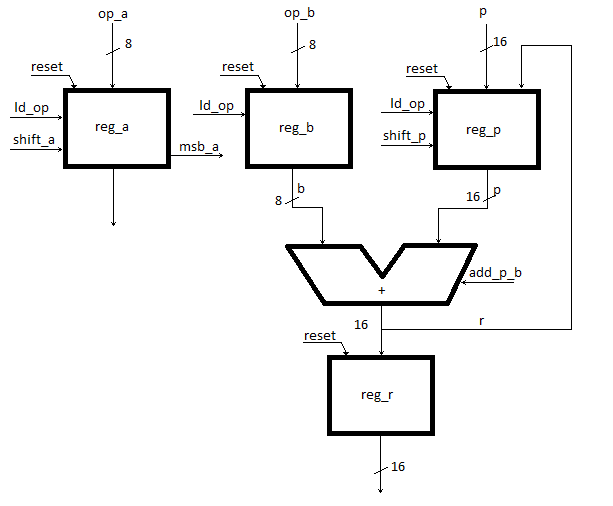
Descrierea sistemului se va face in intregime in VHDL (vbe, fsm). In descrierea de nivel inalt se vor instantia doua componente:

* cale de date
* cale de control

**Reprezentare grafica a structurii de nivel inalt**

****

**Reprezentare grafica a căii de date**

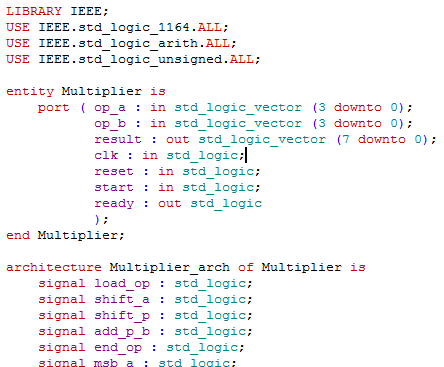


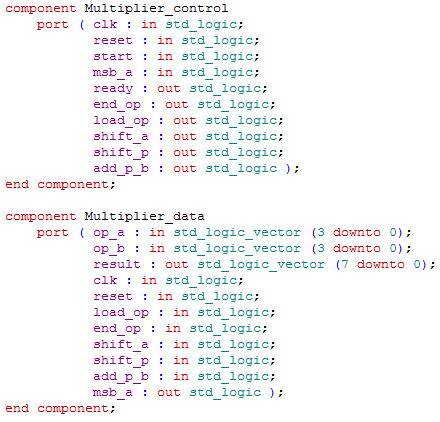
**Tabel cu denumirile si semnificațiile semnalelor interne**

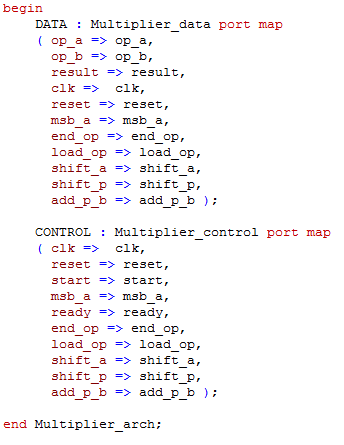
|  |  |
| --- | --- |
| **Denumiresemnal intern** | **Semnificatiile** |
| op\_a | Valoarea care este incarcata in registrul reg\_a |
| op\_b | Valoarea care este incarcata in registrul reg\_b |
| result | Valoarea rezultatului citita din registrul reg\_r |
| clk | Semnalul de ceas |
| reset | Semnal active pe ‘0’ cu rol de trecere a sistemului in starea intiala |
| load\_op | Semnal de incarcare a valorii in registrele reg\_a, reg\_b, reg\_p, reg\_r |
| end\_op | Semnal de finalizare a operatiei de inmultire |
| shift\_a | Semnal de comanda a rotirii valorii la stanga a registrului reg\_a |
| shift\_p | Semnal de comanda a rotirii valorii la stanga a registrului reg\_p |
| add\_p\_b | Semnal de comanda a adunarii in registrul reg\_p a valorii registrului reg\_b |
| msb\_a | Valoareacelui mai semnificativ bit din registrul reg\_a |
| a | Semnal de prelucrare a semnalului op\_a |
| b | Semnal de prelucrare a semnalului op\_b |
| p | Semnal de prelucrare a semnalului op\_p |
| r | Semnal prelucrat r este preluat de result |
| start | Semnal activ in ‘1’ cu rol de trecere in starea SETUP |
| ready | Semnal care comanda finalizarea corecta a operatiei finale |
| curenta | Are ca valoare starea curenta in care se afla sistemul |
| urmatoare | Are ca valoare starea viitoare in care trece sistemul |
| valid | Semnalul prelucrat care este trimis la ready |
| count | Numarul de iteratii pe care le face sistemul |

* Fisierele text sursa **vhdl** cu descrierea sistemului.

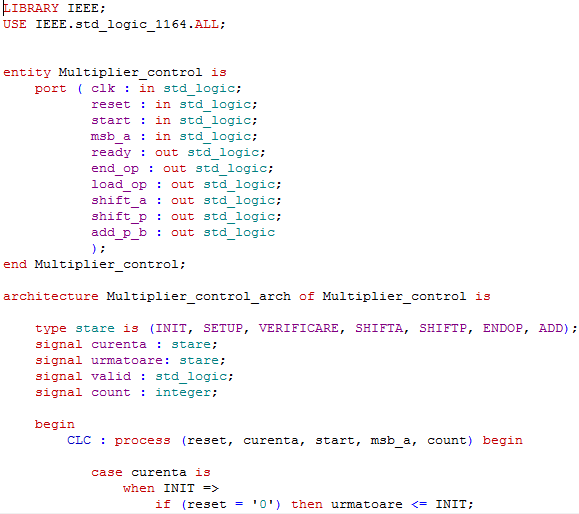
*Multiplier.vhd*

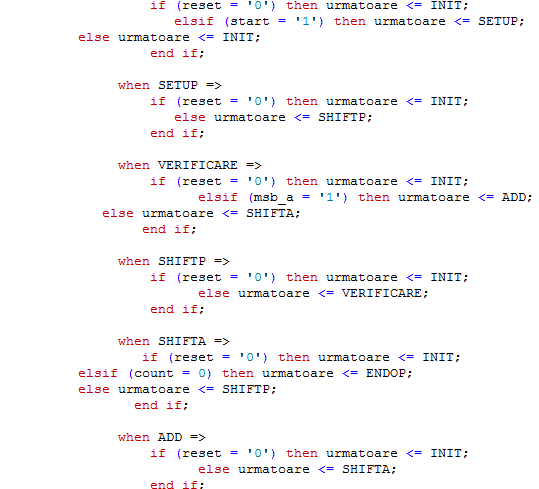


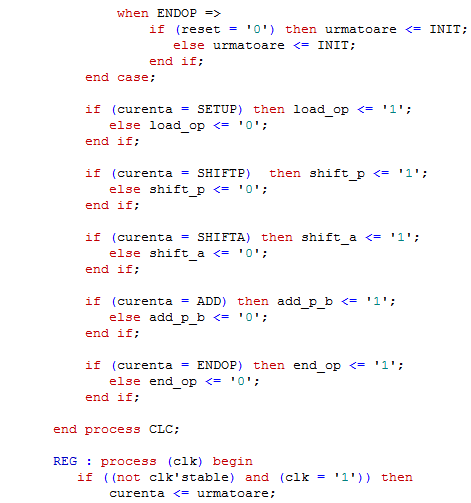


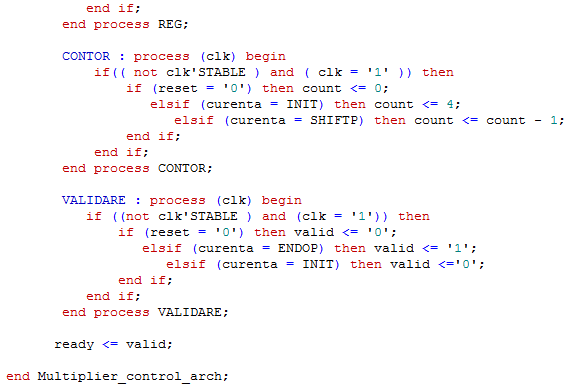


*Multiplier\_control.vhd*

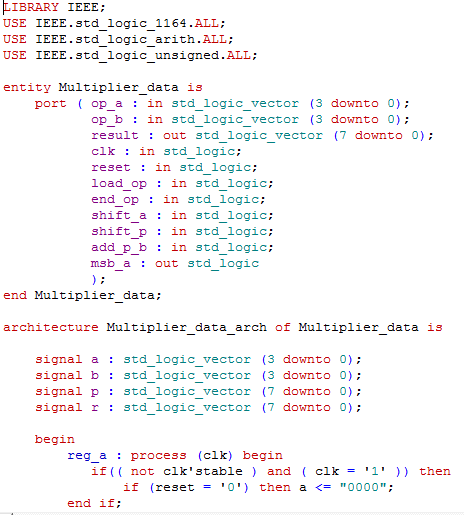


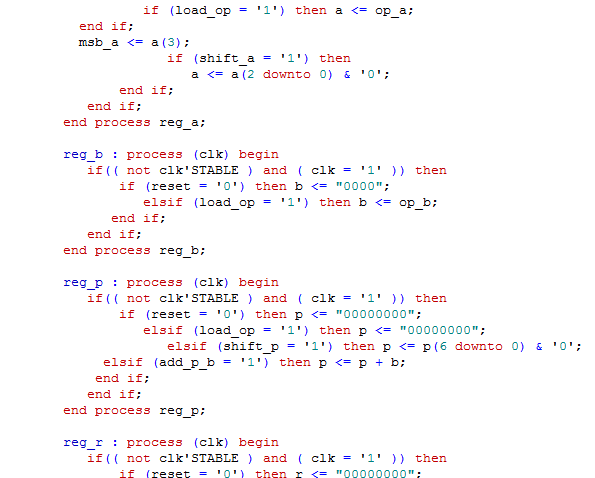


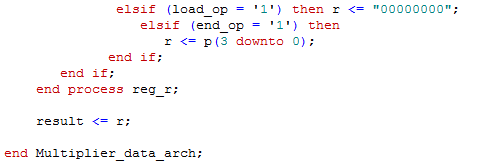




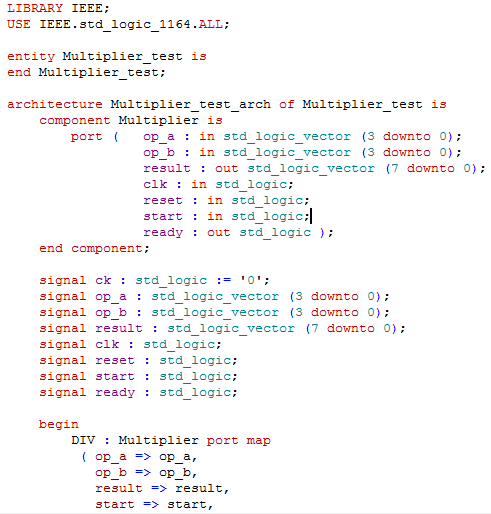
*Multiplier\_data.vhd*

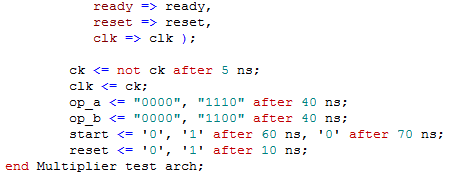






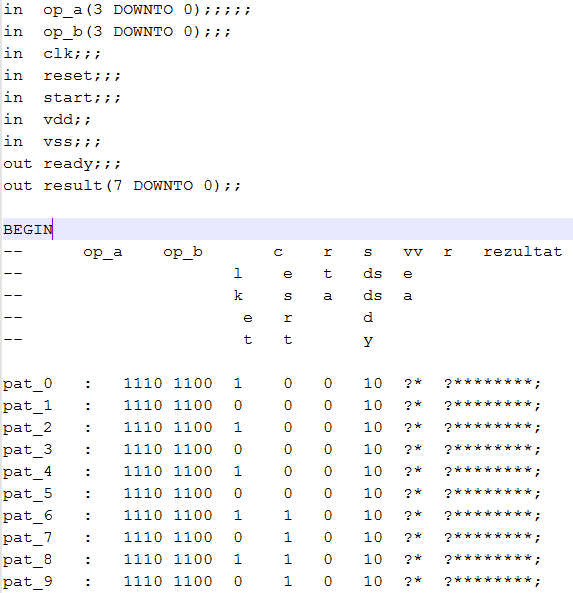
*Multiplier\_test.vhd*



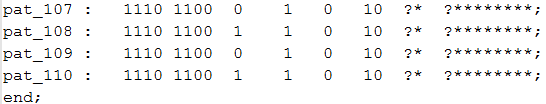


* Fisierele text sursa **pat**folosite la simulare.

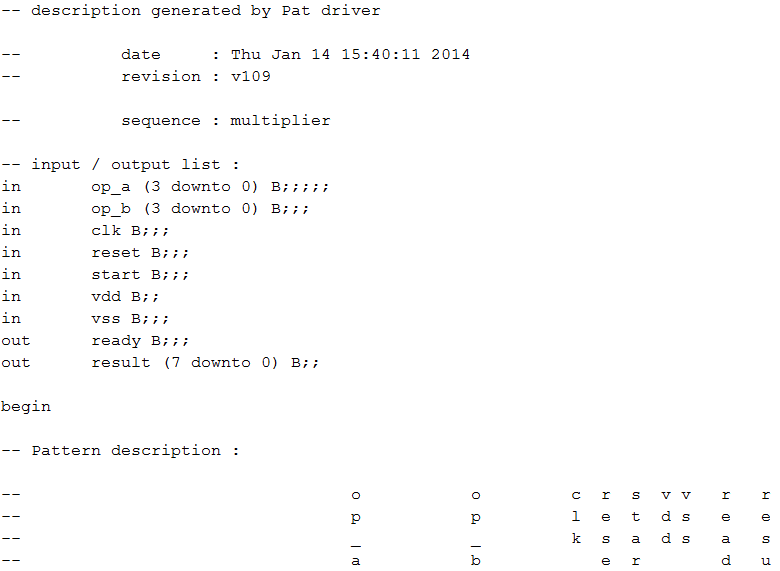
*multiplier.pat*

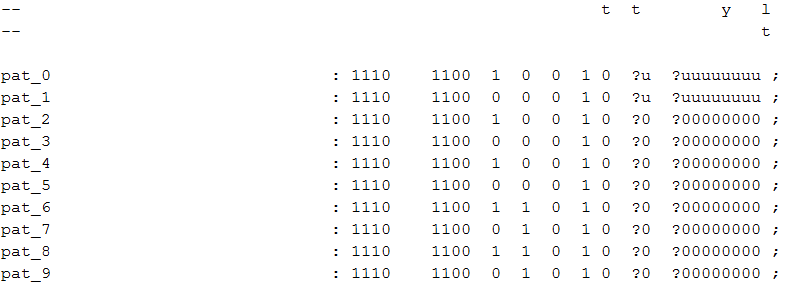


...

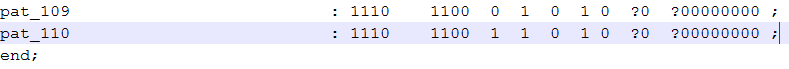


*r1.pat*



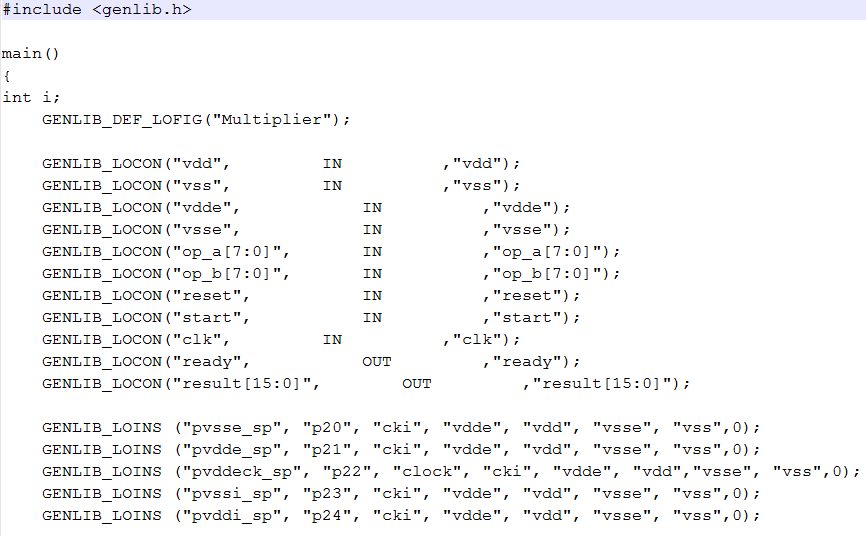


*…*



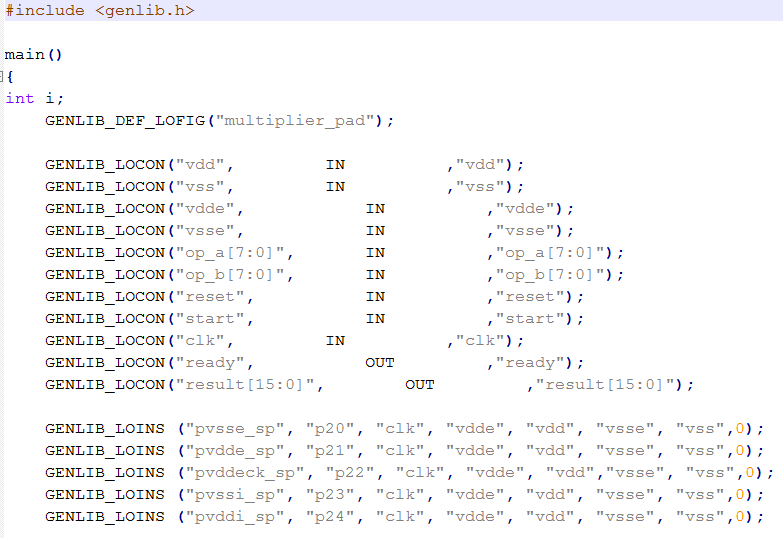
* Fisirele sursa **c**, **rin**reprezentind descrierea structurala, netlist, inclusiv cu asocierea pad-urilor

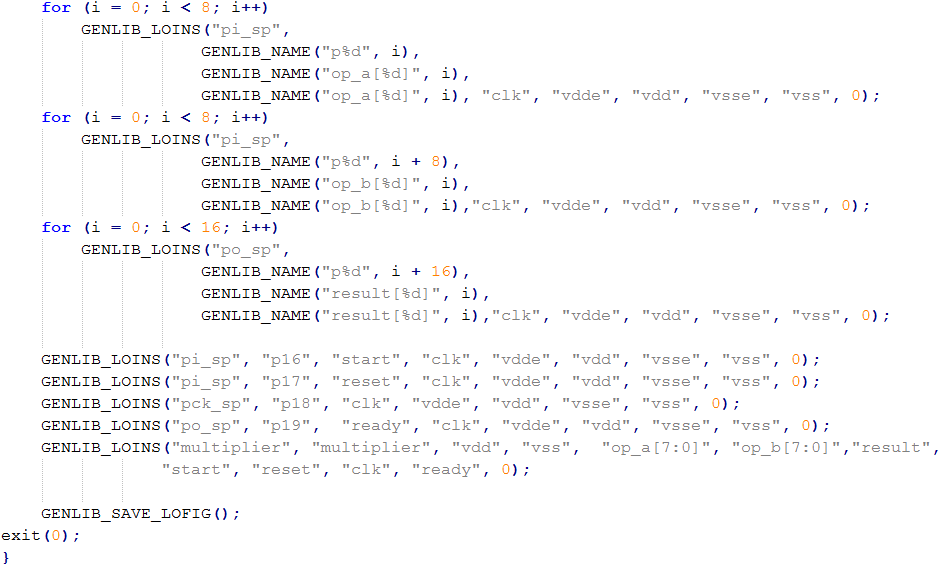
*multiplier.c*





*multiplier\_pad.c*





*multiplier\_pad.c~*

