üáõÝÏóÇ³Ý»ñ Ù ³éç³¹ñ³ÝùÝ» ñ (task)

üáõÝÏóÇ³Ý»ñÁ Ù ³é³ç³¹ñ³ÝùÝ»ñ

- üáõÝÏóÇ³Ý»ñÁ Ù ³é³ç³¹ñ³ÝùÝ»ñÁ (task) ì»ñÇÉá·áõÙ ÑÝ³ñ³íáñáõÃÛáõÝ »Ý ï³ÉÇë ÙÇÙÝáõÛÝ ÁÝÃ³ó³Ï³ñ·Á Ï³ï³ñ»É Íñ³·ñÇ ï³ñµ»ñ ï»Õ»ñÇó:
- ¸ñ³Ýù Ý³Ù ÑÝ³ñ³íáñáõÃÛáõÝ »Ý ï³ÉÇë Ù»Í ·áñÍÁÝÃ³óÝ»ñÁ µ³Å³Ý»É ³í»ÉÇ ųháùñ Ù³ë» ñÇ, ÇÝãÁ å³ñ½»óÝáõÙ ¿ Íñ³·ñÇ ÁÝÃ»éÝ»ÉÇáõÃÛáõÝÁ Ù Ï³ñ·³µ»ñáõÙÁ (ÇÝãå»ë » ÝÃ³Íñ³·ñ»ñÁ):
- Øáõïù/»Éù ³ñ·áõÙ»ÝïÝ»ñÇ ÙÇçáóáí ïíÛ³ÉÝ»ñÁ Ï³ñ»ÉÇ ¿ ųháË³Ýó»É ÁÝÃ³ó³Ï³ñ·ÇÝ Ù ³Û¹ï»ÕÇó Ñ»ï ųháË³Ýó»É Ï³ÝãáÕ Íñ³·ñÇÝ:

- ²é³ç³¹ñ³ÝùÝ»ñÁ Ù ýáõÝÏóÇ³Ý»ñÁ ï³ñµ»ñíáõÙ »Ý Ñ»ïÙÛ³É Ï³ÝáÝÝ»ñáí.
  - üáõÝÏóÇ³Ý å»ïù ¿ Ï³ï³ñíÇ ÝÙ³Ý³ÏÙ³Ý Ù»Ï Å³Ù³Ý³Ï³ÛÇÝ ÙÇ³íáñáõÙ, ³é³ç³¹ñ³ÝùÁ Ï³ñáÕ ¿ å³ñáõÝ³Ï»É Å³Ù³Ý³ÏÇ Ï³é³í³ñÙ³Ý Ï³éáõóí³ÍùÝ»ñ:
  - üáõÝÏóÇ³Ý Ï³ñáÕ ¿ Ï³Ýã»É ýáõÝÏóÇ³, ãÇ Ï³ñáÕ Ï³Ýã»É ³é³ç³¹ñ³Ýù: ²é³ç³¹ñ³ÝùÁ Ï³ñáÕ ¿ Ï³Ýã»É ³ÛÉ ³é³ç³¹ñ³ÝùÝ»ñ Ù ýáõÝÏóÇ³Ý»ñ:
  - üáõÝÏóÇ³Ý å»ïù ¿ áõÝ»Ý³ ³Ù»Ý³ùÇãÁ Ù»Ï ÙáõïùÇ ³ñ·áõÙ»Ýï: ²é³ç³¹ñ³ÝùÁ Ï³ñáÕ ¿ áõÝ» Ý³É ó³ÝÏ³ó³Í ÃíáíÁ ³ñ·áõÙ»ÝïÝ»ñ Ï³Ù Ï³ñáÕ ¿ ³ñ·áõÙ»Ýï ãáõÝ»Ý³É:
  - üáõÝÏóÇ³Ý í»ñ³¹³ñÓÝáõÙ ¿ Ù»Ï ³ñÅ»ù: ²é³ç³¹ñ³ÝùÝ ³ñÅ»ùÝ»ñ ãÇ í»ñ³¹³ñÓÝáõÙ:

²é³ç³¹ñ³Ýù (task)

- ²é³ç³¹ñ³ÝùÇ ë³ÑÙ³ÝáõÙÁ ëÏëíáõÙ ¿ **task** µ³Ý³ÉÇ µ³éáí, áñÇÝ Ï³ñáÕ ¿ Ñ»ïև»É áã å³ñï³¹Çñ **automatic** µ³Ý³ÉÇ µ³éÁ, áñÇÝ Ñ³çáñ¹áõÙ ¿ ³é³ç³¹ñ³ÝùÇ ³Ýí³ÝáõÙÁ և Ï»ï-ëïáñ³Ï»ï:
- ²é³ç³¹ñ³ÝùÇ ë³ÑÙ³ÝáõÙÝ ³í³ñïíáõÙ ¿ **endtask** µ³Ý³ÉÇ µ³éáí:
- **automatic** µ³Ý³ÉÇ µ³éáí Ñ³ÛïÏ³ñ³ñíáõÙ ¿ í»ñ³Ï³ÝãíáÕ ³é³ç³¹ñ³Ýù, áñÇ µáÉáñ Ï³Ýã»ñÇ ïíÛ³ÉÝ»ñÁ ÑÇßáÕáõÃÛáõÝáõÙ å³ÑíáõÙ »Ý ¹ÇÝ³ÙÇÏ Ï»ñåáí:
- ²é³ç³¹ñ³ÝùÇ ë³ÑÙ³ÝáõÙÁ Ï³ñáÕ ¿ å³ñáõÝ³Ï»É Ñ»ïևÛ³É Ñ³Ûï³ñ³ñáõÃÛáõÝÝ»ñÁ`

— **input** ³ñ·áõÙ»ÝïÝ»ñ

— **output** ³ñ·áõÙ»ÝïÝ»ñ

— **inout** ³ñ·áõÙ»ÝïÝ»ñ

— ÁÝÃ³ó³Ï³ñ·³ÛÇÝ µÉáÏáõÙ Ñ³Ûï³ñ³ñíáÕ µáÉáñ ïÇåÇ ïíÛ³ÉÝ»ñ:

•²Ûë Ñ³Ûï³ñ³ñáõÃÛáõÝÝ»ñÇÝ Ñ³çáñ¹áõÙ ¿ ³é³ç³¹ñ³ÝùÇ Ù³ñÙÇÝÁ` Ññ³Ù³ÝÝ»ñÇ Ñ³çáñ¹³Ï³ÝáõÃÛáõÝÁ:

•²é³Ýó **automatic** µ³Ý³ÉÇ µ³éÇ ³é³ç³¹ñ³ÝùÁ ëï³ïÇÏ ¿` µáÉáñ Ñ³Ûï³ñ³ñí³Í ïíÛ³ÉÝ»ñÁ ÑÇßáÕáõÃÛáõÝáõÙ å³ÑíáõÙ »Ý ëï³ïÇÏ Ï»ñåáí: ²Û¹ ïíÛ³ÉÝ»ñÁ Ïû·ï³·áñÍí»Ý ³é³ç³¹ñ³ÝùÇ µáÉáñ ÙÇ³Å³Ù³Ý³Ï Ï³ï³ñíáÕ Ï³Ýã»ñáõÙ: **automatic** ³é³ç³¹ñ³ÝùÇ ïíÛ³ÉÝ»ñÁ ÑÇßáÕáõÃÛáõÝáõÙ ï» Õ³µ³ßËíáõÙ »Ý ¹ÇÝ³ÙÇÏ Ï»ñåáíª Ûáõñ³ù³ÝãÛáõñ Ï³ÝãÇ Ñ³Ù³ñ: ²ÛÝ ³é³ç³¹ñ³ÝùÝ»ñÁ, áñáÝù í» ñ³Ï³ÝãíáÕ »Ý, å»ïù ¿ Ñ³Ûï³ñ³ñí»Ý áñå»ë **automatic**, áñå»ë½Ç ³é³ç³¹ñ³ÝùÇ ÙÇ³Å³Ù³ÏÛ³ Ï³Ýã»ñÁ ã³Õ³í³Õ»Ý ÙÇÙÛ³Ýó ïíÛ³ÉÝ»ñÁ: **automatic** ³é³ç³¹ñ³ÝùÁ å»ïù ¿ Ï³ÝãíÇ Çñ ÑÇ»ñ³ñËÇÏ ³Ýí³ÝáõÙáí:

Ա²é³ç³¹ñ³ÝùÇ ë³ÑÙ³Ýnւմ և Ï³Ýã 1

- ²é³ç³¹ñ³ÝùÁ Ï³ÝãáÕ Ññ³Ù³ÝÁ å»ïù ¿ Ùáõïù/»Éù ³ñ·áõÙ»ÝïÝ»ñÁ վáË³ÝóÇ վ³Ï³·Í»ñáõÙ ëïáñ³Ï» ïáí µ³Å³Ýí³Í ³ñï³Ñ³ÛïáõÃÛáõÝÝ»ñÇ óáõó³ÏÇ ï»ëùáí: ²é³ç³¹ñ³ÝùÇ Ï³ÝãÇ Óև³ã³վÁ Ñ»ïևÛ³ÉÝ ¿.

<name_of_task> (<expression1>,<expression2>,…) ;

áñï»Õ <name_of_task>-Á ³é³ç³¹ñ³ÝùÇ ³Ýí³ÝáõÙÝ ¿, <expression1>, <expression2>,… Ùáõïù/»Éù ³ñ·áõÙ»ÝïÝ»ñÝ »Ý:

- ºÃ» ³é³ç³¹ñ³ÝùÇ ë³ÑÙ³ÝáõÙÁ ³ñ·áõÙ»ÝïÝ»ñ ãáõÝÇ, ³ÛÝ Ï³ÝãíáõÙ ¿ Ýß»Éáí ÙÇ³ÛÝ ³ÝáõÝÁ.

<name_of_task> ;

- ²ñ·áõÙ»ÝïÝ»ñÇ óáõó³ÏÁ å»ïù ÉÇÝÇ ÝáõÛÝ Ñ³çáñ¹Ù³Ý Ï³ñ·áí, ÇÝãå»ë ¹ñ³Ýù Ñ³Ûï³ñ³ñí»É »Ý ³é³ç³¹ñ³ÝùÇ ë³ÑÙ³ÝÙ³Ý Ù»ç:
- Ð»ïևÛ³É ûñÇÝ³ÏÁ óáõó³¹ñáõÙ ¿ ³é³ç³¹ñ³ÝùÇ ë³ÑÙ³ÝÙ³Ý և Ï³ÝãÇ Ï³½Ù³Ï»ñåáõÙÁ.

task my_task;
input a, b;
inout c;
output d, e;

Ué³ç³¹ñ³ÝùÇ ë³ÑÙ³ÝnLU U Ï³Ýã 2

begin // ³Ûëï»Õ ï»Õ³¹ñíáõÙ »Ý ³é³ç³¹ñ³ÝùÇ ³ßË³ï³ÝùÝ Çñ³Ï³ÝóÝáÕ Ññ³Ù³ÝÝ»ñÁ
c = foo1; // »ÉùÇ ³ñ·áõÙ»ÝïÝ»ñÇÝ í»ñ³·ñ»É
d = foo2; // ³é³ç³¹ñ³ÝùÇ ³ßË³ï³ÝùÇ ³ñ¹ÛáõÝùÝ»ñÁ
e = foo3;
end
endtask

Ð»ïUÛ³É Ññ³Ù³Ýáí Ï³ÝãíáõÙ ¿ my_task ³é³ç³¹ñ³ÝùÁ`
my_task (v, w, x, y, z);
Î³ÝãÇ (v, w, x, y, z) ³ñ·áõÙ»ÝïÝ»ñÁ Ñ³Ù³å³ï³ëË³ÝáõÙ »Ý (a, b, c, d, e) Ùáõïù/»Éù ³ñ·áõÙ»ÝïÝ»ñÇÝ, áñáÝù Ñ³ÛïñÍ³ñ³ñí»É »Ý ³é³ç³¹ñ³ÝùÇ ë³ÑÙ³ÝÙ³Ý Ù»ç: ²é³ç³¹ñ³ÝùÇ Ï³ÝãÇ Å³Ù³Ý³Ï **input** U **inout** ïÇåÇ (a, b, c) ³ñ·áõÙ» ÝïÝ»ñÁ ëï³ÝáõÙ »Ý v, w, x ųáųáË³Ï³ÝÝ»ñÇ ³ñÅ»ùÝ»ñÁ:
a = v;
b = w;
c = x;
²é³ç³¹ñ³ÝùÇ Ï³ï³ñÙ³Ý ³ñ¹ÛáõÝùÝ»ñÁ ï»Õ³¹ñíáõÙ »Ý **output** U **inout** ïÇåÇ c, d, e ųáųáË³Ï³ÝÝ»ñáõÙ: ºñµ ³é³ç³¹ñ³ÝùÇ Ï³ï³ñáõÙÝ ³í³ñïíáõÙ ¿, Íñ³·ñÇ Ï³é³í³ñáõÙÁ í»ñ³¹³ñÓíáõÙ ¿ Ï³ÝãáÕ Íñ³·ñÇÝ, áñÁ c, d, e ųáųáË³Ï³ÝÝ»ñáí ëï³óí³Í ³ñ¹ÛáõÝùÝ»ñÁ í»ñ³·ñáõÙ ¿ Ï³ÝãáÕ Íñ³·ñÇ x, y, z ųáųáË³Ï³ÝÝ»ñÇÝ`
x = c;
y = d;
z = e;

Առաջադրանքի օրինակ (շարունակելի)

```
module traffic_lights;
reg clock, red, amber, green;//ï³Ïï³ÛÇÝ ÇÙåáõÉëÝ»ñÇ, Ï³ñÙÇñ, ¹»ÕÇÝ և Ï³Ý³ã ·áõÛÝ»ñÇ
                                //ÉáõÛë»ñÇ íÇß³ÏÝ»ñÇ é»·ÇëïñÝ»ñ
parameter on = 1, off = 0;     // on-ÙÇ³ó³Í íÇ×³Ï, off-³Ýç³ïí³Í íÇß³Ï
parameter red_tics = 350, amber_tics = 30, green_tics = 200; //ÉáõÛë»ñÇ ïևáÕáõÃÛáõÝÁ
                                                              //ë³ÑÙ³ÝáÕ å³ñ³Ù»ïñ»ñ
initial red = off;                  // ï³É ÉáõÛë»ñÇ ëÏ½µÝ³Ï³Ý íÇ×³ÏÝ»ñÁ
initial amber = off;
initial green = off;
always
begin                          // ÉáõÛë»ñÇ Ï³é³í³ñÙ³Ý Ñ³çáñ¹³Ï³ÝáõÃÛáõÝÁ.
      red = on;                // ÙÇ³óÝ»É Ï³ñÙÇñ ÉáõÛëÁ
      light(red, red_tics);     //ëå³ë»É` Ï³Ýã»É light ³é³ç³¹ñ³ÝùÁ
      green = on;              // ÙÇ³óÝ»É Ï³Ý³ã ÉáõÛëÁ
      light(green, green_tics); // ëå³ë»É
      amber = on;              // ÙÇ³óÝ»É ¹»ÕÇÝ ÉáõÛëÁ
      light(amber, amber_tics); // ëå³ë»É
end
```

Առաջադրանքի օրինակ (շարունակություն)

```
// light ³é³ç³¹ñ³ÝùÇ ë³ÑÙ³ÝáõÙ: ²Ûë ³é³ç³¹ñ³Ýùáí Óև³íáñíáõÙ ¿ Å³Ù³Ý³Ï³ÛÇÝ //Ñ³å³ÕáõÙ, áñÇ
ïևáÕáõÃÛáõÝÁ Ñ³í³ë³ñ ¿  clock ÇÙåáõÉëÝ»ñÇ tics Ãíáí //å³ñµ»ñáõÃÛáõÝÝ»ñÇ: ²Û¹
Ñ³å³ÕÙ³Ý ³í³ñïÇÝ ÉáõÛëÝ ³Ýç³ïíáõÙ ¿:
task light;
output color;
input [31:0] tics;
begin
repeat (tics)              //ÏñÏÝ»É tics ³Ý·³Ù
   @ (posedge clock);
   color = off;            // ³Ýç³ï»É ÉáõÛëÁ
end
endtask

always                     // clock ï³Ïï³ÛÇÝ ³½¹³Ýß³ÝÇ Óև³íáñáõÙ
begin
      #100 clock = 0;
      #100 clock = 1;
end
endmodule      // traffic_lights.
```

Վերլոգ ֆունկցիաներ

- üáõÝÏóÇ³Ý í»ñ³¹³ñÓÝáõÙ ¿ Ù»Ï ³ñÅ»ù, áñÁ Ï³ñáÕ ¿ û·ï³·áñÍí»É ³ñï³Ñ³Ûï áõÃÛáõÝÝ»ñáõÙ: üáõÝÏóÇ³Ý Ñ³Ûï³ñ³ñíáõÙ ¿ **function** և **endfunction** µ³Ý³ÉÇ µ³é»ñáí: üáõÝÏóÇ³Ý áõÝÇ ³í»ÉÇ ë³ÑÙ³Ý³ψ³Ï ÑÝ³ñ³íáñáõÃÛáõÝÝ»ñ, ù³Ý ³é³ç³¹ñ³ÝùÁ: üáõÝÏóÇ³Ý å»ïù ¿ µ³í³ñ³ñÇ Ñ»ïևÛ³É Ï³ÝáÝÝ»ñÇÝ.

•1. üáõÝÏóÇ³Ý ãÇ Ï³ñáÕ å³ñáõÝ³Ï»É Å³Ù³Ý³ÏÇ Ï³é³í³ñÙ³Ý áñ¨¿ Ññ³Ù³Ý: ²ÛëÇÝùÝ ýáõÝÏóÇ³ÛáõÙ ã»Ý Ï³ñáÕ û·ï³·áñÍí»É՝ **#, @, Ï³Ù wait** ëÇÙíáÉÝ»ñ Ï³Ù Ï³éáõóí³ÍùÝ»ñ:

•2. üáõÝÏóÇ³Ý ãÇ Ï³ñáÕ Ï³Ýã»É ³é³ç³¹ñ³Ýù, ³ÛÝ Ï³ñáÕ ¿ Ï³Ýã»É ³ÛÉ ýáõÝÏóÇ³Ý»ñ:

•3. üáõÝÏóÇ³Ý å»ïù ¿ áõÝ»Ý³ ³éÝí³½Ý Ù»Ï ÙáõïùÇ ³ñ·áõÙ»Ýï՝ **input** ïÇåÇ:

•4. üáõÝÏóÇ³ÛÇ ë³ÑÙ³ÝáõÙÁ å»ïù ¿ Ý»ñ³éÇ ýáõÝÏóÇ³ÛÇ ³ñ¹ÛáõÝùÁ ýáõÝÏóÇ³ÛÇ ³ÝáõÝÁ ÏñáÕ Ý» ñùÇÝ ÷á÷áË³Ï³ÝÇÝ í»ñ³·ñÙ³Ý Ññ³Ù³Ý:

- **üáõÝÏóÇ³ÛÇ ë³ÑÙ³ÝÙ³Ý** ÁÝ¹Ñ³Ýáõñ Ï³éáõóí³ÍùÁ Ñ»ïևÛ³ÉÝ ¿.

function <range_or_type> <name_of_function> ;

<tf_declaration> //÷á÷áË³Ï³ÝÝ»ñÇ Ñ³Ûï³ñ³ñáõÙ

<statement> //ýáõÝÏóÇ³ÛÇ ³ßË³ï³ÝùÁ áñáßáÕ Ññ³Ù³ÝÝ»ñÁ

endfunction

- ²Ûëï»Õ՝ <range_or_type> ¹³ßïáõÙ ïñíáõÙ ¿ ýáõÝÏóÇ³Ûáí í»ñ³¹³ñÓíáÕ ψáψáË³Ï³ÝÇ µÇÃ»ñÇ ïÇñáõÛÃÁ ([n:m] ï»ëùáí) և ïÇåÁ, »Ã» ³ÛÝ **integer** Ï³Ù ¿ **real** ¿: ºÃ» <range_or_type> ¹³ßïÁ µ³ó³Ï³ÛáõÙ ¿, ³å³ ýáõÝÏóÇ³Ûáí í»ñ³¹³ñÓíáÕ ψáψáË³Ï³ÝÁ 1-µÇÃ é»·Çëïñ ¿: <name_of_function> ýáõÝÏóÇ³ÛÇ ³ÝáõÝÝ ¿՝ Ç¹»ÝïÇýÇÏ³ïáñÁ: úñÇÝ³Ï,

function [7:0] adder ; // adder ³Ýí³ÝáõÙáí 8-µÇÃ é»·Çëïñ ïÇåÇ:

Ֆnւնկgհայի ÙáõïùÇ և Ý»ñùÇÝ վáվáË³Ï³ÝÝ»ñը

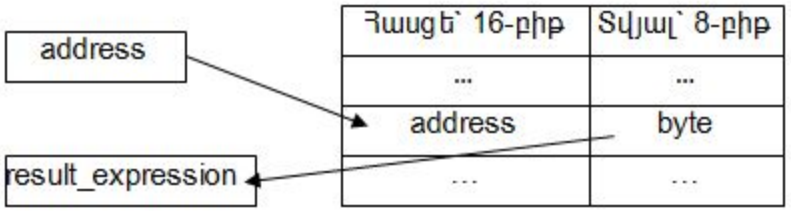
<tf_declaration> ÁÝ¹·ñÏáõÙ ¿ ÙáõïùÇ և Ý»ñùÇÝ վáվáË³Ï³ÝÝ»ñÇ, å³ï³Ñ³ñÝ»ñÇ և å³ñ³Ù» ïñ»ñÇ Ñ³Ûï³ñ³ñáõÙ
<parameter_declaration>;
<input_declaration> ;
<output_declaration> ;
<inout_declaration> ;
<reg_declaration>;
<time_declaration>;
<integer_declaration>;
<real_declaration>;
<event_declaration>;

²Ûëï»Õ Ãí³ñÏí³ÍÝ»ñÇó å³ñï³¹Çñ ¿ ÙÇ³ÛÝ <input_declaration>-Á:

Ð»ïևÛ³É ûñÇÝ³ÏáõÙ ë³ÑÙ³ÝíáõÙ ¿ getbyte ³Ýí³ÝáõÙáí ýáõÝÏóÇ³.

function [7:0] getbyte;
input [15:0] address;
begin
<statements> // Ññ³Ù³ÝÝ»ñ, áñáÝóáí address Ñ³ëó»Çó ïíÛ³ÉÇ µ³ÛÃÁ ÁÝÃ»ñóíáõÙ ÑÇßáÕáõÃÛáõÝÇó
getbyte = result_expression; //³ñ¹ÛáõÝùÁ í»ñ³·ñíáõÙ ¿ ýáõÝóÇ³ÛÇ
//³ÝáõÝÁ ÏñáÕ վհáՎհáËªï³ÝÇÝ

end
endfunction



üáõÝÏóÇ³ÛÇ ë³ÑÙ³ÝáõÙÁ

- üáõÝÏóÇ³ÛÇ ë³ÑÙ³ÝáõÙÁ ³Ýµ³ó³Ñ³ÛÏ Ï»ñåáí Ñ³ÛÏ³ñ³ñáõÙ ¿ Ý»ñùÇÝ é»·Çïñ, áñÝ áõÝÇ ÝáõÛÝ ³Ýí³ÝáõÙÁ, ÇÝã ýáõÝÏóÇ³Ý: ²Ûë é»·ÇëïñÇ ã³÷ëÁ 1-µÇÃ ¿ Ï³Ù ïñíáõÙ ¿ ýáõÝÏóÇ³ÛÇ Ñ³ÛÏ³ñ³ñÙ³Ý <range_or_type> ¹³ßïáõÙ: üáõÝÏóÇ³ÛÇ ë³ÑÙ³ÝÙ³Ý Ù»ç å»ïù ¿ ÉÇÝÇ ýáõÝÏóÇ³ÛÇ ³ßË³ï³ÝùÇ ³ñ¹ÛáõÝùÁ ýáõÝÏóÇ³ÛÇ ³ÝáõÝÁ ÏñáÕ Ý»ñùÇÝ ųháųháË³Ï³ÝÇÝ í»ñ³·ñ»Éáõ Ññ³Ù³Ý`

getbyte = result_expression;

- **üáõÝÏóÇ³ÛÇ Ï³ÝãÝ** Çñ³Ï³Ý³óíáõÙ ¿ Ñ»ïłÛ³É Ï³éáõóí³Ílùáí.

<name_of_function> (<expression>,<expression>,…)

üáõÝÏóÇ³ÛÇ Ï³ÝãÁ Ï³ñáÕ ¿ ÉÇÝ»É áñł¿ ³ñï³Ñ³ÛïáõÃÛ³Ý ûå»ñ³ïáñ: Ð»ïłÛ³É ûñÇÝ³ÏáõÙ word ³ñï³Ñ³ÛïáõÃÛáõÝáõÙ getbyte ýáõÝÏóÇ³ÛÇ »ñÏáõ Ï³Ýã»ñÇó Ół³íáñíáõÙ ¿ 16-µÇÃ µ³é` µ³ÛÃ»ñÇ ÙÇ³Ïóáõ Ùáí.

word = control ? {getbyte(msbyte), getbyte(lsbyte)} : 0;// msbyte ł lsbyte ³ñ¹ÛáõÝ³ñ³ñ

//µ³éÇ ³í³· ł Ïñïë»ñ µ³ÛÃ»ñÇ Ñ³ëó»Ý»ñÝ »Ý:

Ֆունկցիայի կիրառության օրինակներ

- //ê³ÑÙ³Ý»É Ùá¹áõÉ, áñÁ å³ñáõÝ³ÏáõÙ ¿ even_parity ³Ýí³ÝáõÙáí ýáõÝÏóÇ³

```
module check_parity;
...
reg [63:0] data;
reg parity; //Ñ³ßí»É 64-µÇÃ ïíÛ³ÉáõÙ ½áõÛ·áõÃÛ³Ý Ñ³ÛÏ³ÝÇßÁ, »ñµ ³Û¹ ïíÛ³ÉÁ ÷áËíáõÙ ¿
always @(data)
begin
parity = even_parity(data);    //Ï³Ýã»É even_parity ýáõÝÏóÇ³Ý
$display("Parity status = %b", parity);
end
.....
//ë³ÑÙ³Ý»É even_parity ýáõÝÏóÇ³Ý
function  even_parity; //ýáõÝÏóÇ³ÛÇ ³ßË³ï³ÝùÇ ³ñ¹ÛáõÝùÁ í»ñ³¹³ñÓíáõÙ ¿  even_parity                //
³Ýµ³ó³Ñ³Ûï Ñ³Ûï³ñ³ñíáÕ é»·Çëïñáí
input [63:0] data;
begin
even_parity = ^ data; //Ï³ï³ñ»É ë»ÕÙÙ³Ý xor ·áñÍáÕáõÃÛáõÝ data-Ç µáÉáñ µÇÃ»ñÇ ÝÏ³ïÙ³Ùµ
endendfunction          //ýáõÝÏóÇ³ÛÇ ë³ÑÙ³ÝÙ³Ý ³í³ñï.....
endmodule        // check_parity Ùá¹áõÉÇ ³í³ñï
```

Տվյալների գտում

Î³½Ù»É ì»ñÇÉá· Ùá¹áõÉª ïíÛ³ÉÝ»ñÇ ½ïÙ³Ý Ñ³Ù³ñ: ê³ÑÙ³Ý»É ýáõÝÏóÇ³ª Ñ»ï¨Û³É »ñÏñáñ¹ Ï³ñ·Ç ½ïÇãÇ Çñ³Ï³Ý³óÙ³Ý Ñ³Ù³ñ.

y(n)=0.25x(n)+0.5x(n-1)+0.25x(n-2):

Øáõïù³ÛÇÝ x և »Éù³ÛÇÝ y ïíÛ³ÉÝ»ñÁ 8-µÇÃ »Ý:

```
module fir3(clk,x,y);
input clk;
input [7:0] x;
output [7:0] y;
reg [7:0] x1, x2;

always @(posedge clk)
begin
        x1<=x; // x ïíÛ³ÉÇ Ñ³å³ÕáõÙ Ù»Ï ï³Ïïáí` x1=x(n-1)
        x2<=x1;          //x2=x(n-2)
end

assign y=fir3(x, x1, x2);

function [7:0] fir3;
input [7:0] a, b, c;
assign fir3= (a >> 2) + (b >> 1) + (c >> 2) ;
endfunction

endmodule
```