

üáõÝİóÇ³Ý»ñ ı ³éç³¹ñ³ÝùÝ»
ñ (task)

üáõÝíóÇ³Ý»ñÁ ì ³é³ç³¹ñ³ÝùÝ»ñ

- üáõÝíóÇ³Ý»ñÁ ì ³é³ç³¹ñ³ÝùÝ»ñÁ (task) ì»ñÇÉá·áõÙ ÑÝ³ñ³íañáõÃÛáõÝ »Ý ï³ÉÇë ÙÇìÝáõÛÝ ÁÝÃ³ó³í³ñ·Á ï³í³ñ»É Ìñ³·ñÇ ï³ñμ»ñ ï»Õ»ñÇó:
- ,ñ³Ýù Ý³ì ÑÝ³ñ³íañáõÃÛáõÝ »Ý ï³ÉÇë Ù»Í ·áñÍÁÝÃ³óÝ»ñÁ μ³Á³Ý»É ³í»ÉÇ ψáùñ Ù³ë» ñÇ, ÇÝãÁ á³ñ½»óÝáõÙ ¿ Ìñ³·ñÇ ÁÝÃ»éÝ»ÉÇáõÃÛáõÝÁ ì ï³ñ·³μ»ñáõÛÁ (ÇÝãá»ë » ÝÃ³íñ³·ñ»ñÁ):
- Øáõïù/»Éù ³ñ·áõÙ»ÝíÝ»ñÇ ÙÇÇáóái ïïÛ³ÉÝ»ñÁ ï³ñ»ÉÇ ¿ ψáË³Ýó»É ÁÝÃ³ó³í³ñ·ÇÝ ì ³Û¹ï»ÕÇó Ñ»ï ψáË³Ýó»É ï³ÝááÕ Ìñ³·ñÇÝ:
- ²é³ç³¹ñ³ÝùÝ»ñÁ ì ýáõÝíóÇ³Ý»ñÁ ï³ñμ»ñíaõÙ »Ý Ñ»ïìùÛ³É ï³ÝáÝÝ»ñái.
 - üáõÝíóÇ³Ý á»ïù ¿ ï³í³ñíÇ Ýù³Ý³ïÛ³Ý Ù»Í Á³ù³Ý³ïÛÇÝ ÙÇ³íañáõÙ, ³é³ç³¹ñ³ÝùÁ ï³ñáÕ ¿ á³ñáõÝ³í»É Á³ù³Ý³íÇ ï³é³í³ñÛ³Ý ï³éáóóí³íùÝ»ñ:
 - üáõÝíóÇ³Ý ï³ñáÕ ¿ ï³Ýã»É ýáõÝíóÇ³, áÇ ï³ñáÕ ï³Ýã»É ³é³ç³¹ñ³Ýù: ²é³ç³¹ñ³ÝùÁ ï³ñáÕ ¿ ï³Ýã»É ³ÛÉ ³é³ç³¹ñ³ÝùÝ»ñ ì ýáõÝíóÇ³Ý»ñ:
 - üáõÝíóÇ³Ý á»ïù ¿ áõÝ»Ý³ ³ù»Ý³ùÇãÁ Ù»Í ÙáõïùÇ ³ñ·áõÙ»Ýí: ²é³ç³¹ñ³ÝùÁ ï³ñáÕ ¿ áõÝ»Ý³É ó³Ýí³ó³í Áíái ³ñ·áõÙ»ÝíÝ»ñ ï³Û ï³ñáÕ ¿ ³ñ·áõÙ»Ýí ááõÝ»Ý³É:
 - üáõÝíóÇ³Ý í»ñ³¹³ñÓÝáõÙ ¿ Ù»Í ³ñÁ»ù: ²é³ç³¹ñ³ÝùÝ ³ñÁ»ùÝ»ñ áÇ í»ñ³¹³ñÓÝáõÙ:

²é³ç³¹ñ³Ýù (task)

- ²é³ç³¹ñ³ÝùÇ ë³ÑÙ³ÝáðÙÁ ëĭĕiáðÙ ħ **task** μ³Ý³ÉÇ μ³éáí, áñÇÝ ĩ³ñáÕ ħ Ñ»ĭlu»É áã á³ñĩ³¹Çñ **automatic** μ³Ý³ÉÇ μ³éÁ, áñÇÝ Ñ³çáñ¹áðÙ ħ ³é³ç³¹ñ³ÝùÇ ³Ýí³ÝáðÙÁ ĭ Ĩ»ĩ-ĕiáñ³Ĭ»ĩ:
- ²é³ç³¹ñ³ÝùÇ ë³ÑÙ³ÝáðÙÝ ³ĩ³ñĩiáðÙ ħ **endtask** μ³Ý³ÉÇ μ³éáí:
- **automatic** μ³Ý³ÉÇ μ³éáí Ñ³Ùĩ³ñ³ñiáðÙ ħ ĩ»ñ³ĩ³ÝáñáÕ ³é³ç³¹ñ³Ýù, áñÇ μáÉáñ ĩ³Ýã»ñÇ ĩĩÙ³ÉÝ»ñÁ ÑÇáÕáðÁÙáðÝáðÙ á³ÑiáðÙ »Ý ¹ÇÝ³ÙÇĩ Ĩ»ñáí:
- ²é³ç³¹ñ³ÝùÇ ë³ÑÙ³ÝáðÙÁ ĩ³ñáÕ ħ á³ñáðÝ³Ĭ»É Ñ»ĭluÙ³É Ñ³Ùĩ³ñ³ñáðÁÙáðÝÝ»ñÁ`

— **input** ³ñ·áðÙ»ÝĩÝ»ñ

— **output** ³ñ·áðÙ»ÝĩÝ»ñ

— **inout** ³ñ·áðÙ»ÝĩÝ»ñ

— ÁÝÃ³óĩ³ñ·³ÙÇÝ μÉáíáðÙ Ñ³Ùĩ³ñ³ñiáðÙ μáÉáñ ĩÇáÇ ĩĩÙ³ÉÝ»ñ:

•²Ùë Ñ³Ùĩ³ñ³ñáðÁÙáðÝÝ»ñÇÝ Ñ³çáñ¹áðÙ ħ ³é³ç³¹ñ³ÝùÇ Ù³ñÙÇÝÁ` Ññ³Ù³ÝÝ»ñÇ Ñ³çáñ¹³ĩ³ÝáðÁÙáðÝÁ:

•²é³Ýó **automatic** μ³Ý³ÉÇ μ³éÇ ³é³ç³¹ñ³ÝùÁ ĕĩ³ĩÇĩ ħ` μáÉáñ Ñ³Ùĩ³ñ³ñiáðÙ ĩĩÙ³ÉÝ»ñÁ ÑÇáÕáðÁÙáðÝáðÙ á³ÑiáðÙ »Ý ĕĩ³ĩÇĩ Ĩ»ñáí: ²Ù¹ ĩĩÙ³ÉÝ»ñÁ ĩù·ĩ³·áñĩ»Ý ³é³ç³¹ñ³ÝùÇ μáÉáñ ÙÇ³Á³Ù³Ý³ĩ ĩ³ñiáðÙ ĩ³Ýã»ñáðÙ: **automatic** ³é³ç³¹ñ³ÝùÇ ĩĩÙ³ÉÝ»ñÁ ÑÇáÕáðÁÙáðÝáðÙ ĩ» Õ³μ³ßĕiáðÙ »Ý ¹ÇÝ³ÙÇĩ Ĩ»ñáíª Ùáðñ³ù³ÝãÙáðñ ĩ³ÝãÇ Ñ³Ù³ñ: ²ÙÝ ³é³ç³¹ñ³ÝùÝ»ñÁ, áñáÝù ĩ» ñ³ĩ³ÝáñáÕ »Ý, á»ĭù ħ Ñ³Ùĩ³ñ³ñiáðÙÝ áñá»ë **automatic**, áñá»ë½Ç ³é³ç³¹ñ³ÝùÇ ÙÇ³Á³Ù³ĩÙ³ ĩ³Ýã»ñÁ á³Õ³ĩ³Õ»Ý ÙÇÙÙ³Ýó ĩĩÙ³ÉÝ»ñÁ: **automatic** ³é³ç³¹ñ³ÝùÁ á»ĭù ħ ĩ³ÝáĩÇ Çñ ÑÇ»ññĕÇĩ ³Ýí³ÝáðÙáí:

Ué³ç³¹ñ³ÝùÇ ë³ÑÛ³ÝñŁ u İ³Ýã 1

- ²é³ç³¹ñ³ÝùÁ İ³ÝãáÕ Ññ³Û³ÝÁ á»İù ĸ Ùáõİù/»Éù ³ñ·áõÛ»ÝİÝ»ñÁ ĸáË³ÝóÇ ĸ³İ³·Í»ñáõÛ ëİáñ³İ» İáÍ μ³Á³ÝİÍ ³ñİ³Ñ³ÛİáõÃÛáõÝÝ»ñÇ óáõó³İÇ İ»ëùáÍ: ²é³ç³¹ñ³ÝùÇ İ³ÝãÇ Ół³ã³ĸÁ Ñ»İłÛ³ÉÝ ĸ.

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

áñİ»Õ *<name_of_task>*-Á ³é³ç³¹ñ³ÝùÇ ³Ýİ³ÝáõÛÝ ĸ, *<expression1>*, *<expression2>*, ... Ùáõİù/»Éù ³ñ·áõÛ»ÝİÝ»ñÝ »Ý:

- °Ã» ³é³ç³¹ñ³ÝùÇ ë³ÑÛ³ÝáõÛÁ ³ñ·áõÛ»ÝİÝ»ñ ááõÝÇ, ³ÛÝ İ³ÝáíáõÛ ĸ Ýß»ÉáÍ ÛÇ³ÛÝ ³ÝáõÝÁ.

<name_of_task> ;

- ²ñ·áõÛ»ÝİÝ»ñÇ óáõó³İÁ á»İù ÉÇÝÇ ÝáõÛÝ Ñ³çáñ¹Û³Ý İ³ñ·áÍ, ÇÝã»ë ¹ñ³Ýù Ñ³Ûİ³ñ³ñİ»É »Ý ³é³ç³¹ñ³ÝùÇ ë³ÑÛ³ÝÛ³Ý Û»ç:
- Ð»İłÛ³É ûñÇÝ³İÁ óáõó³¹ñáõÛ ĸ ³é³ç³¹ñ³ÝùÇ ë³ÑÛ³ÝÛ³Ý u İ³ÝãÇ İ³¹²Û³İ»ñááõÛÁ.

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

Ué³ç³¹ñ³ÝùÇ ë³ÑÛ³ÝnŁ u İ³Ýã 2

```
begin // ³Ûëï»Õ ï»Õ³¹ñíáðÛ »Ý ³é³ç³¹ñ³ÝùÇ ³ßË³í³ÝùÝ Çñ³í³ÝóÝáÕ Ññ³Û³ÝÝ»ñÁ
c = foo1; // »ÉùÇ ³ñ·áðÛ»ÝíÝ»ñÇÝ í»ñ³·ñ»É
d = foo2; // ³é³ç³¹ñ³ÝùÇ ³ßË³í³ÝùÇ ³ñ¹ÛáðÝùÝ»ñÁ
e = foo3;
end
endtask
```

Đ»İlÛ³É Ññ³Û³Ýáí İ³ÝãíáðÛ ĸ my_task ³é³ç³¹ñ³ÝùÁ`
my_task (v, w, x, y, z);
İ³ÝãÇ (v, w, x, y, z) ³ñ·áðÛ»ÝíÝ»ñÁ Ñ³Û³á³³ëË³ÝáðÛ »Ý (a, b, c, d, e) Ûáðïù»Éù ³ñ·áðÛ»ÝíÝ»ñÇÝ, áñáÝù Ñ³Û³ñ³ñí»É »Ý ³é³ç³¹ñ³ÝùÇ ë³ÑÛ³ÝÛ³Ý Û»Ç: ²é³ç³¹ñ³ÝùÇ İ³ÝãÇ Á³Û³Ýİ **input** u **inout** ĩÇâÇ (a, b, c) ³ñ·áðÛ»ÝíÝ»ñÁ ëí³ÝáðÛ »Ý v, w, x ¼á¼áË³İ³ÝÝ»ñÇ ³ñÁ»ùÝ»ñÁ:
a = v;
b = w;
c = x;
²é³ç³¹ñ³ÝùÇ İ³í³ñÛ³Ý ³ñ¹ÛáðÝùÝ»ñÁ ï»Õ³¹ñíáðÛ »Ý **output** u **inout** ĩÇâÇ c, d, e ¼á¼áË³İ³ÝÝ»ñáðÛ: °ñµ ³é³ç³¹ñ³ÝùÇ İ³í³ñáðÛÝ ³í³ñííáðÛ ĸ, Íñ³·ñÇ İ³é³í³ñáðÛÁ í»ñ³¹³ñÓíáðÛ ĸ İ³ÝãáÕ Íñ³·ñÇÝ, áñÁ c, d, e ¼á¼áË³İ³ÝÝ»ñáí ëí³óíí ³ñ¹ÛáðÝùÝ»ñÁ í»ñ³·ñáðÛ ĸ İ³ÝãáÕ Íñ³·ñÇ x, y, z ¼á¼áË³İ³ÝÝ»ñÇÝ`
x = c;
y = d;
z = e;

Առաջադրանքի օրինակ (շարունակելի)

```
module traffic_lights;
reg clock, red, amber, green; // i³i³ûçý çùááõéëý»ñç, i³ñùçñ, 1»õçý և i³ý³ã ·áõûý»ñç
// Éáõûë»ñç íçđ³íý»ñç é»·çëiñý»ñ
parameter on = 1, off = 0; // on-ùç³ó³í íç×³í, off-³ýç³í³í íçđ³í
parameter red_tics = 350, amber_tics = 30, green_tics = 200; // Éáõûë»ñç iláõáõãûáõýá
// ë³ñû³ýáõ ă³ñ³ù»iñ»ñ
initial red = off; // i³É Éáõûë»ñç ëi½μý³í³ý íç×³íý»ñÁ
initial amber = off;
initial green = off;
always
begin // Éáõûë»ñç i³é³i³ñû³ý Ñ³çáñ¹³i³ýáõãûáõýá.
    red = on; // ùç³óý»É i³ñùçñ ÉáõûëÁ
    light(red, red_tics); // ëá³ë»É` i³ýã»É light ³é³ç³¹ñ³ýùÁ
    green = on; // ùç³óý»É i³ý³ã ÉáõûëÁ
    light(green, green_tics); // ëá³ë»É
    amber = on; // ùç³óý»É 1»õçý ÉáõûëÁ
    light(amber, amber_tics); // ëá³ë»É
end
```

Առաջադրանքի օրինակ (շարունակություն)

```
// light ³é³ç³¹ñ³ÝùÇ ë³ÑÛ³ÝáðÙ: ²Ûë ³é³ç³¹ñ³Ýùáí ÓÙ³íáñíáðÙ ¿ Á³Û³Ý³í³ÛÇÝ //Ñ³á³ÕáðÙ, áñÇ  
ííáÕáðÃÛáðÝÁ Ñ³í³ë³ñ ¿ clock ÇÙááðÉëÝ»ñÇ tics Áíáí //á³ñµ»ñáðÃÛáðÝÝ»ñÇ: ²Û¹  
Ñ³á³ÕÛ³Ý ³í³ñíÇÝ ÉáðÛëÝ ³Ýç³íáðÙ ¿:
```

```
task light;
```

```
output color;
```

```
input [31:0] tics;
```

```
begin
```

```
repeat (tics) //íñíÝ»É tics ³Ý·³Û
```

```
  @ (posedge clock);
```

```
  color = off; // ³Ýç³í»É ÉáðÛëÁ
```

```
end
```

```
endtask
```

```
always // clock íñí³ÛÇÝ ³½¹³Ýß³ÝÇ ÓÙ³íáñáðÙ
```

```
begin
```

```
  #100 clock = 0;
```

```
  #100 clock = 1;
```

```
end
```

```
endmodule // traffic_lights.
```

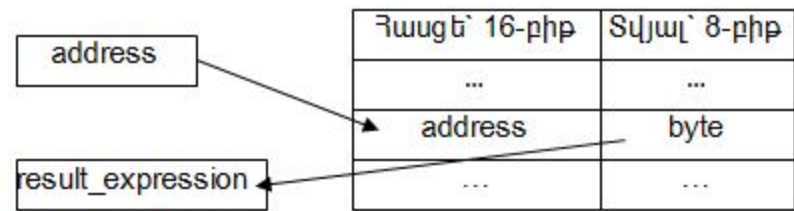

Συντακτική Σύμβαση και Συμβολισμός της VHDL

<tf_declaration> Αίτιο-αποτέλεσμα, Σύμβαση και Συμβολισμός της VHDL, Αίτιο-αποτέλεσμα και Αίτιο-αποτέλεσμα

- <parameter_declaration>;
- <input_declaration>;
- <output_declaration>;
- <inout_declaration>;
- <reg_declaration>;
- <time_declaration>;
- <integer_declaration>;
- <real_declaration>;
- <event_declaration>;

Αίτιο-αποτέλεσμα Αίτιο-αποτέλεσμα και Σύμβαση και Συμβολισμός της VHDL, Αίτιο-αποτέλεσμα και Αίτιο-αποτέλεσμα

```
function [7:0] getbyte;
input [15:0] address;
begin
<statements> // Αίτιο-αποτέλεσμα, Αίτιο-αποτέλεσμα address Αίτιο-αποτέλεσμα
    getbyte = result_expression; // Αίτιο-αποτέλεσμα και Αίτιο-αποτέλεσμα
end
endfunction
```



üáõÝĭóÇ³ÛÇ ë³ÑÛ³ÝáõÙÁ

- üáõÝĭóÇ³ÛÇ ë³ÑÛ³ÝáõÙÁ ³Ýµ³ó³Ñ³Ûĭ ĩ»ñááĭ Ñ³Ûĭ³ñ³ñáõÙ ħ Ý»ñùÇÝ é»·Çĭñ, áñÝ áõÝÇ ÝáõÛÝ ³Ýĭ³ÝáõÙÁ, ÇÝã ýáõÝĭóÇ³Ý: ²Ûë é»·ÇëĭñÇ ã³÷ëÁ 1-µÇÃ ħ ĩ³Û ĩñíaõÙ ħ ýáõÝĭóÇ³ÛÇ Ñ³Ûĭ³ñ³ñÛ³Ý <range_or_type> ¹³ĬíaõÙ: üáõÝĭóÇ³ÛÇ ë³ÑÛ³ÝÛ³Ý Û»Ç á»ĭù ħ ÉÇÝÇ ýáõÝĭóÇ³ÛÇ ³ßË³ĭ³ÝùÇ ³ñ¹ÛáõÝùÁ ýáõÝĭóÇ³ÛÇ ³ÝáõÝÁ ĩñáÕ Ý»ñùÇÝ ŸáŸáË³ĭ³ÝÇÝ ĩ»ñ³·ñ»Éáõ Ññ³Û³Ý`

getbyte = result_expression;

- **üáõÝĭóÇ³ÛÇ ĩ³ÝãÝ Çñ³ĭ³Ý³óíaõÙ ħ Ñ»ĭĭÛ³É ĩ³éáõóĭùáí.**
<name_of_function> (<expression>,<expression>,...)

üáõÝĭóÇ³ÛÇ ĩ³ÝãÁ ĩ³ñáÕ ħ ÉÇÝ»É áñĭ ħ ³ñĭ³Ñ³ÛĭíaõÃÛ³Ý ùá»ñ³íañ: Ð»ĭĭÛ³É ùñÇÝ³íaõÙ word ³ñĭ³Ñ³ÛĭíaõÃÛáõÝáõÙ getbyte ýáõÝĭóÇ³ÛÇ »ñíaõ ĩ³Ýã»ñÇó Óĭ³íañíaõÙ ħ 16-µÇÃ µ³é` µ³ÛÃ»ñÇ ÛÇ³íaõõÙáí.

word = control ? {getbyte(msbyte), getbyte(lsbyte)} : 0;// msbyte ĭ lsbyte ³ñ¹ÛáõÝ³ñ³ñ
//µ³éÇ ³ĭ³· ĭ ĩñĭë»ñ µ³ÛÃ»ñÇ Ñ³ëó»Ý»ñÝ »Ý:

Տվյալների գտում

Ինքնուրույն լինելու պայմաններում $y(n) = 0.25x(n) + 0.5x(n-1) + 0.25x(n-2)$ հարաբերակցությունը կարող է իրականացվել հետևյալ կոդով:

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

Քանի որ x և y ունենում են 8-բիտ լայնություն:

```
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 նախորդ քայլի արժեքը
    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
```