

üáõÝİóÇ³Ý»ñ ı ³éç³¹ñ³ÝùÝ»  
ñ (task)

# üáõÝíóÇ³Ý»ñÁ Ì ³é³ç³¹ñ³ÝùÝ»ñ

- üáõÝíóÇ³Ý»ñÁ Ì ³é³ç³¹ñ³ÝùÝ»ñÁ (task) Ì»ñÇÉá·áõÙ ÑÝ³ñ³íáñáõÃÙáõÝ »Ý Ì³ÉÇë ÙÇÌÝáõÙÝ ÁÝÃ³ó³í³ñ·Á Ì³í³ñ»É Ìñ³·ñÇ Ì³ñµ»ñ Ì»Õ»ñÇó:
- ,ñ³Ýù Ý³Ì ÑÝ³ñ³íáñáõÃÙáõÝ »Ý Ì³ÉÇë Ù»Ì ·áñÍÁÝÃ³óÝ»ñÁ µ³Á³Ý»É ³í»ÉÇ ÷áùñ Ù³ë» ñÇ, ÇÝãÁ á³ñ½»óÝáõÙ ¿ Ìñ³·ñÇ ÁÝÃ»éÝ»ÉÇáõÃÙáõÝÁ Ì Ì³ñ·³µ»ñáõÙÁ (ÇÝãá»ë » ÝÃ³íñ³·ñ»ñÁ):
- Øáõìù/»Éù ³ñ·áõÙ»ÝíÝ»ñÇ ÙÇÇáóái ÌÛ³ÉÝ»ñÁ Ì³ñ»ÉÇ ¿ ÷áË³Ýó»É ÁÝÃ³ó³í³ñ·ÇÝ Ì ³Ù¹ì»ÕÇó Ñ»ì ÷áË³Ýó»É Ì³ÝãáÕ Ìñ³·ñÇÝ:
- ²é³ç³¹ñ³ÝùÝ»ñÁ Ì ýáõÝíóÇ³Ý»ñÁ Ì³ñµ»ñíáõÙ »Ý Ñ»ìÌÙ³É Ì³ÝáÝÝ»ñái.
  - üáõÝíóÇ³Ý á»ìù ¿ Ì³í³ñíÇ ÝÙ³Ý³ÌÙ³Ý Ù»Ì Á³Ù³Ý³ÌÙÇÝ ÙÇ³íáñáõÙ, ³é³ç³¹ñ³ÝùÁ Ì³ñáÕ ¿ á³ñáõÝ³Í»É Á³Ù³Ý³ÌÇ Ì³é³í³ñÙ³Ý Ì³éáóóí³ÌùÝ»ñ:
  - üáõÝíóÇ³Ý Ì³ñáÕ ¿ Ì³Ýã»É ýáõÝíóÇ³, áÇ Ì³ñáÕ Ì³Ýã»É ³é³ç³¹ñ³Ýù: ²é³ç³¹ñ³ÝùÁ Ì³ñáÕ ¿ Ì³Ýã»É ³ÙÉ ³é³ç³¹ñ³ÝùÝ»ñ Ì ýáõÝíóÇ³»ñ:
  - üáõÝíóÇ³Ý á»ìù ¿ áõÝ»Ý³ ³Ù»Ý³ùÇãÁ Ù»Ì ÙáõìùÇ ³ñ·áõÙ»Ýí: ²é³ç³¹ñ³ÝùÁ Ì³ñáÕ ¿ áõÝ»Ý³É ó³Ýí³ó³í Áíái ³ñ·áõÙ»ÝíÝ»ñ Ì³Ù Ì³ñáÕ ¿ ³ñ·áõÙ»Ýí ááõÝ»Ý³É:
  - üáõÝíóÇ³Ý í»ñ³¹³ñÓÝáõÙ ¿ Ù»Ì ³ñÁ»ù: ²é³ç³¹ñ³ÝùÝ ³ñÁ»ùÝ»ñ áÇ í»ñ³¹³ñÓÝáõÙ:

# ²é³ç³¹ñ³Ýù (task)

- ²é³ç³¹ñ³ÝùÇ ë³ÑÙ³ÝáðÙÁ ëĭĕĭáðÙ ħ **task** μ³Ý³ÉÇ μ³éáĭ, áñÇÝ ĭ³ñáŎ ħ Ñ»ĭĭ»É áā á³ñĭ³¹Çñ **automatic** μ³Ý³ÉÇ μ³éÁ, áñÇÝ Ñ³çáñ¹áðÙ ħ ³é³ç³¹ñ³ÝùÇ ³Ýĭ³ÝáðÙÁ ĭ ĭ»ĭ-ĕĭáñ³ĭĭ:
- ²é³ç³¹ñ³ÝùÇ ë³ÑÙ³ÝáðÙÝ ³ĭñĭĭáðÙ ħ **endtask** μ³Ý³ÉÇ μ³éáĭ:
- **automatic** μ³Ý³ÉÇ μ³éáĭ Ñ³Ùĭ³ñ³ñĭáðÙ ħ ĭ»ñ³ĭ³ÝáĭáŎ ³é³ç³¹ñ³Ýù, áñÇ μáÉáñ ĭ³Ýā»ñÇ ĭĭÙ³ÉÝ»ñÁ ÑÇáŎáðĀÙáðÝáðÙ á³ÑĭáðÙ »Ý ¹ÇÝ³ÙÇĭ ĭ»ñááĭ:
- ²é³ç³¹ñ³ÝùÇ ë³ÑÙ³ÝáðÙÁ ĭ³ñáŎ ħ á³ñáðÝ³ĭ»É Ñ»ĭĭĭÙ³É Ñ³Ùĭ³ñ³ñáðĀÙáðÝÝ»ñÁ`

— **input** ³ñ·áðÙ»ÝĭÝ»ñ

— **output** ³ñ·áðÙ»ÝĭÝ»ñ

— **inout** ³ñ·áðÙ»ÝĭÝ»ñ

— ÁÝĀ³ó³ĭ³ñ·³ÙÇÝ μÉáĭáðÙ Ñ³Ùĭ³ñ³ñĭáŎ μáÉáñ ĭÇáÇ ĭĭÙ³ÉÝ»ñ:

•²Ùë Ñ³Ùĭ³ñ³ñáðĀÙáðÝÝ»ñÇÝ Ñ³çáñ¹áðÙ ħ ³é³ç³¹ñ³ÝùÇ Ù³ñÙÇÝÁ` Ññ³Ù³ÝÝ»ñÇ Ñ³çáñ¹³ĭ³ÝáðĀÙáðÝÁ:

•²é³Ýó **automatic** μ³Ý³ÉÇ μ³éÇ ³é³ç³¹ñ³ÝùÁ ĕĭĭĭÇĭ ħ` μáÉáñ Ñ³Ùĭ³ñ³ñĭáĭ ĭĭÙ³ÉÝ»ñÁ ÑÇáŎáðĀÙáðÝáðÙ á³ÑĭáðÙ »Ý ĕĭĭĭÇĭ ĭ»ñááĭ: ²Ù¹ ĭĭÙ³ÉÝ»ñÁ ĭù·ĭ³·áñĭĭ»Ý ³é³ç³¹ñ³ÝùÇ μáÉáñ ÙÇ³Á³Ù³Ý³ĭ ĭ³ñĭáŎ ĭ³Ýā»ñáðÙ: **automatic** ³é³ç³¹ñ³ÝùÇ ĭĭÙ³ÉÝ»ñÁ ÑÇáŎáðĀÙáðÝáðÙ ĭ» Ŏ³μ³ßĕĭáðÙ »Ý ¹ÇÝ³ÙÇĭ ĭ»ñááĭá Ùáðñ³ù³ÝāÙáðñ ĭ³ÝāÇ Ñ³Ù³ñ: ²ÙÝ ³é³ç³¹ñ³ÝùÝ»ñÁ, áñáÝù ĭ» ñ³ĭ³ÝáĭáŎ »Ý, á»ĭù ħ Ñ³Ùĭ³ñ³ñĭ»Ý áñá»ë **automatic**, áñá»ë½Ç ³é³ç³¹ñ³ÝùÇ ÙÇ³Á³Ù³ĭÙ³ ĭ³Ýā»ñÁ á³Ŏ³ĭ³Ŏ»Ý ÙÇÙÙ³Ýó ĭĭÙ³ÉÝ»ñÁ: **automatic** ³é³ç³¹ñ³ÝùÁ á»ĭù ħ ĭ³ÝáĭÇ Çñ ÑÇ»ññĕĭĭ ³Ýĭ³ÝáðÙáĭ:

# 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
```

Ð»ïŁÛ³É 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 Á³Û³Ý³İ **input** u **inout** ĩÇâÇ (a, b, c) ³ñ·áðÛ»ÝíÝ»ñÁ ëí³ÝáðÛ »Ý v, w, x ÷á÷áËË³í³ÝÝ»ñÇ ³ñÁ»ùÝ»ñÁ:  
a = v;  
b = w;  
c = x;  
²é³ç³¹ñ³ÝùÇ İ³í³ñÛ³Ý ³ñ¹ÛáðÝùÝ»ñÁ ï»Õ³¹ñíáðÛ »Ý **output** u **inout** ĩÇâÇ c, d, e ÷á÷áËË³í³ÝÝ»ñáðÛ: °ñµ ³é³ç³¹ñ³ÝùÇ İ³í³ñáðÛÝ ³í³ñííáðÛ ĸ, Íñ³·ñÇ İ³é³í³ñáðÛÁ í»ñ³¹³ñÓíáðÛ ĸ İ³ÝãáÕ Íñ³·ñÇÝ, áñÁ c, d, e ÷á÷áËË³í³ÝÝ»ñáí ëí³óí³í ³ñ¹ÛáðÝùÝ»ñÁ í»ñ³·ñáðÛ ĸ İ³ÝãáÕ Íñ³·ñÇ x, y, z ÷á÷áËË³í³ÝÝ»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 íçð³íý»ñç é»·çëiñý»ñ
parameter on = 1, off = 0; // on-ùç³ó³í íç×³í, off-³ýç³í³í íçð³í
parameter red_tics = 350, amber_tics = 30, green_tics = 200; // Éáõûë»ñç iláõáõãûáõýá
// ë³ñû³ýáõ ã³ñ³ù»iñ»ñ
initial red = off; // i³É Éáõûë»ñç ëi½μý³í³ý íç×³íý»ñÁ
initial amber = off;
initial green = off;
always
begin // Éáõûë»ñç i³é³í³ñû³ý 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 ÓÙ³íañíaðÙ ¿ Å³Û³Ý³i³ÛÇÝ //Ñ³á³ÕáðÙ, áñÇ  
iláÕáðÃÛáðÝÁ Ñ³i³ë³ñ ¿ clock ÇÙááðÉëÝ»ñÇ tics Áíáí //á³ñµ»ñáðÃÛáðÝÝ»ñÇ: ²Û¹  
Ñ³á³ÕÛ³Ý ³i³ñiÇÝ ÉáðÛëÝ ³Ýç³iíaðÙ ¿:
```

```
task light;
```

```
output color;
```

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

```
begin
```

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

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

```
  color = off; // ³Ýç³i»É ÉáðÛëÁ
```

```
end
```

```
endtask
```

```
always // clock i³i³ÛÇÝ ³½¹³Ýß³ÝÇ ÓÙ³íañíaðÙ
```

```
begin
```

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

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

```
end
```

```
endmodule // traffic_lights.
```

# Վերլուծող Ֆունկցիաներ

- `üáðÝíóÇ³Ý` `í»ñ³¹³ñÓÝáðÙ` `¿` `Ù»í` `³ñÀ»ù`, `áñÁ` `í³ñáÕ` `¿` `ù·í³·áñí`»`É` `³ñ³Ñ³ÙíáðÃÙáðÝÝ»ñáðÙ`:  
`üáðÝíóÇ³Ý` `Ñ³Ùí³ñ³ñíáðÙ` `¿` **function** `ü` **endfunction** `µ³Ý³ÉÇ` `µ³é»ñáí`: `üáðÝíóÇ³Ý` `áðÝÇ` `³í»ÉÇ`  
`ë³ÑÙ³Ý³` `í³í` `ÑÝ³ñ³íáñáðÃÙáðÝÝ»ñ`, `ù³Ý` `³é³Ç³¹³ñ³ÝùÁ`: `üáðÝíóÇ³Ý` `á»íù` `¿` `µ³í³ñ³ñÇ` `Ñ»íüÙ³É`  
`í³ÝáÝÝ»ñÇÝ`.

- 1. `üáðÝíóÇ³Ý` `áÇ` `í³ñáÕ` `á³ñáðÝ³í»É` `Á³Ù³Ý³íÇ` `í³é³í³ñÙ³Ý` `áñ` `¿` `Ññ³Ù³Ý`: `²ÙéÇÝùÝ` `ýáðÝíóÇ³ÙáðÙ` `ã»Ý`  
`í³ñáÕ` `ù·í³·áñí`»`É` `#`, `@`, `í³Ù` **wait** `ëÇÙíáÉÝ»ñ` `í³Ù` `í³éáðóí³íùÝ»ñ`:

- 2. `üáðÝíóÇ³Ý` `áÇ` `í³ñáÕ` `í³Ýá»É` `³é³Ç³¹³ñ³Ýù`, `³ÙÝ` `í³ñáÕ` `¿` `í³Ýá»É` `³ÙÉ` `ýáðÝíóÇ³Ý»ñ`:

- 3. `üáðÝíóÇ³Ý` `á»íù` `¿` `áðÝ»Ý³` `³éÝí³½Ý` `Ù»í` `ÙáðíùÇ` `³ñ·áðÙ»Ýí` **input** `íÇáÇ`:

- 4. `üáðÝíóÇ³ÙÇ` `ë³ÑÙ³ÝáðÙÁ` `á»íù` `¿` `Ý»ñ³éÇ` `ýáðÝíóÇ³ÙÇ` `³ñ¹ÙáðÝùÁ` `ýáðÝíóÇ³ÙÇ` `³ÝáðÝÁ` `íñáÕ` `Ý»`  
`ñùÇÝ` `÷á÷áÉ³í³ÝÇÝ` `í»ñ³·ñÙ³Ý` `Ññ³Ù³Ý`:

- **`üáðÝíóÇ³ÙÇ`** `ë³ÑÙ³ÝÙ³Ý` `ÁÝ¹Ñ³Ýáðñ` `í³éáðóí³íùÁ` `Ñ»íüÙ³ÉÝ` `¿`.

**function** `<range_or_type>` `<name_of_function>` ;

`<tf_declaration>` `//÷á÷áÉ³í³ÝÇ»ñÇ` `Ñ³Ùí³ñ³ñíáðÙ`

`<statement>` `//ýáðÝíóÇ³ÙÇ` `³É³¹³ÝùÁ` `áñáááÕ` `Ññ³Ù³ÝÝ»ñÁ`

**endfunction**

- `²Ùëí»Õ` `<range_or_type>` `¹³íáðÙ` `íñíáðÙ` `¿` `ýáðÝíóÇ³Ùáí` `í»ñ³¹³ñÓíáÕ` `íáíáÉ³í³ÝÇ` `µÇÃ»ñÇ`  
`íÇñáðÙÁÁ` (`[n:m]` `í»ëùáí`) `ü` `íÇáÁ`, `»Ã»` `³ÙÝ` **integer** `í³Ù` `¿` **real** `¿`: `°Ã»` `<range_or_type>` `¹³íÁ`  
`µ³ó³í³ÙáðÙ` `¿`, `³á³` `ýáðÝíóÇ³Ùáí` `í»ñ³¹³ñÓíáÕ` `íáíáÉ³í³ÝÁ` `1-µÇÃ` `é»·Çëíñ` `¿`:  
`<name_of_function>` `ýáðÝíóÇ³ÙÇ` `³ÝáðÝÝ` `¿` `Ç¹»ÝíÇÝÇí³íáñÁ`: `úñÇÝ³í`,

**function** `[7:0]` `adder` ; `// adder` `³Ýí³ÝáðÙáí` `8-µÇÃ` `é»·Çëíñ` `íÇáÇ`:



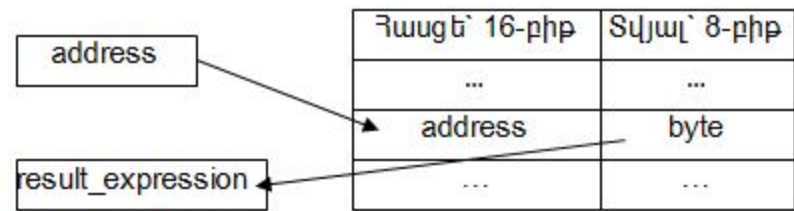
# Συντακτική Σύμβαση και Συμβολισμός της VHDL

**<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 ĭ»ñááĭ Ñ³Ûĭ³ñ³ñáõÙ ħ Ý»ñùÇÝ é»·Çĭñ, áñÝ áõÝÇ ÝáõÛÝ ³Ýĭ³ÝáõÙÁ, ÇÝã ýáõÝĭóÇ³Ý: ²Ûë é»·ÇëĭñÇ ã³÷ëÁ 1-µÇÃ ħ ĭ³Û ĭñĭáõÙ ħ ýáõÝĭóÇ³ÛÇ Ñ³Ûĭ³ñ³ñÛ³Ý <range\_or\_type> ¹³ĭĭáõÙ: üáõÝĭóÇ³ÛÇ ë³ÑÛ³ÝÛ³Ý Û»Ç á»ĭù ħ ÉÇÝÇ ýáõÝĭóÇ³ÛÇ ³ßË³ĭ³ÝùÇ ³ñ¹ÛáõÝùÁ ýáõÝĭóÇ³ÛÇ ³ÝáõÝÁ ĭñáÕ Ý»ñùÇÝ ŸáŸáË³ĭ³ÝÇÝ ĭ»ñ³·ñ»Éáõ Ññ³Û³Ý`

getbyte = result\_expression;

- **üáõÝĭóÇ³ÛÇ ĭ³ÝãÝ Çñ³ĭ³Ý³óĭáõÙ ħ Ñ»ĭĭÛ³É ĭ³éáõóĭáõÙáĭ.**  
<name\_of\_function> ( <expression>,<expression>,... )

üáõÝĭóÇ³ÛÇ ĭ³ÝãÁ ĭ³ñáÕ ħ ÉÇÝ»É áñĭ ħ ³ñĭ³Ñ³ÛĭáõÃÛ³Ý ùá»ñ³ĭáñ: Ð»ĭĭÛ³É ùñÇÝ³ĭáõÙ word ³ñĭ³Ñ³ÛĭáõÃÛáõÝáõÙ getbyte ýáõÝĭóÇ³ÛÇ »ñĭáõ ĭ³Ýã»ñÇó Óĭ³ĭáñĭáõÙ ħ 16-µÇÃ µ³é` µ³ÛÃ»ñÇ ÛÇ³ĭáõÙáĭ.

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

# Ֆունկցիայի կիրառության օրինակներ

```

• //Է՞ՆՍՅՅ»Է Սա՛աճԷ, ա՞ՆԱ օ՞ճՅՅՅՅՅՅՅ շ even_parity ՅՅՅՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅ
module check_parity;
...
reg [63:0] data;
reg parity; //ՆՅՅՅ»Է 64-մՅՅ իՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅՅՅ, »ճմ ՅՅՅՅ իՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅ շ
always @(data)
begin
parity = even_parity(data); //ՅՅՅՅ»Է even_parity ՅՅՅՅՅՅՅՅՅՅ
$display("Parity status = %b", parity);
end

.....
//Է՞ՆՍՅՅ»Է even_parity ՅՅՅՅՅՅՅՅՅՅ
function even_parity; //ՅՅՅՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅՅՅ շ even_parity //
ՅՅՅՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅՅՅ շ
input [63:0] data;
begin
even_parity = ^ data; //ՅՅՅՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅՅՅ xor ՅՅՅՅՅՅՅՅՅՅՅՅՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅՅՅ
endendfunction //ՅՅՅՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅՅՅ ՅՅՅՅՅՅՅՅՅՅՅ
endmodule // check_parity Սա՛աճԷՅՅ ՅՅՅՅՅՅ

```

# Տվյալների գտում

Ինքնուրույն լինելու պայմաններում  $y(n) = 0.25x(n) + 0.5x(n-1) + 0.25x(n-2)$  հարաբերակցությունը կարող է ներկայացվել հետևյալ կերպով՝

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

Քանի որ  $x$  և  $y$  նշանակում են 8-բիտ թվեր, ապա:

```
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 նշանակում է նախորդ քայլի արժեքը 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
```