طراحی کامپیوتری سیستم‌های دیجیتال

جلسه چهارم



بردارها در وری لاگ

در این زبان بردار بیت ها با استفاده ازبراکت [ ] مشخص می شود.

A[15:0] (برداری شامل 16 بیت)

A[6] (بیت ششم ازبردار A)

A[7:0] (هشت بیت اول از بردار A)

مثال:

```
output [0: 3] D;
wire   [7: 0] SUM;
```

$$^{MSB}[R_a:R_b]^{LSB}$$

دسترسی و انتخاب بیت های مورد نظر

```
// An 8-bit vector
reg [7:0] Areg;
```




```
// declaration: reg [7:0]  Areg;
Areg [7:5]
```




```
Areg [5:3]    // selects bits 5, 4, and 3
```

شروع از بیت

```
Areg [5-:4]   // selects bits 5, 4, 3, and 2
```

جهت

تعداد بیت ها

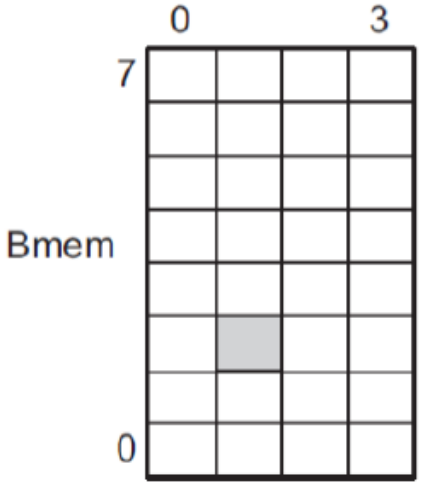
```
Areg [2+:4]   // selects bits 5, 4, 3, and 2
```

حافظه

در وریلاگ به آرایه های چند بعدی حافظه یا memory گفته می شود.

حافظه ها را می توان از نوع reg یا integer تعریف کرد.

```
// declaration: reg Bmem [7:0][0:3]
Bmem [2] [1]
```

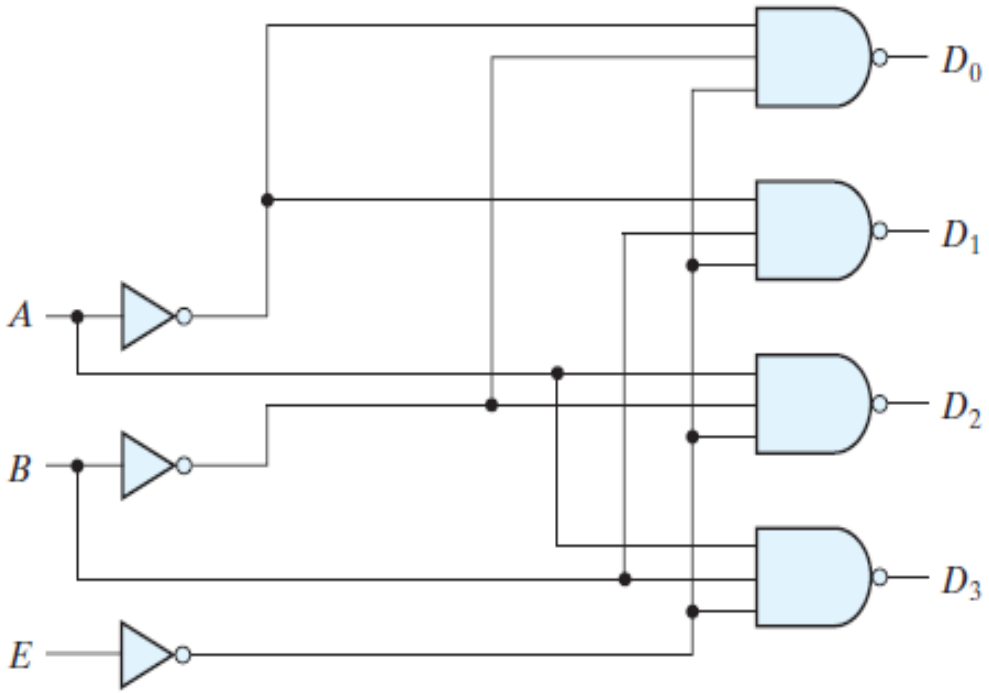


*داده نوع integer حداقل 32 بیت در نظر گرفته می شود ومقادیر علامت دار را شامل می شود.

مثال

کد برگردان دو-به-چهار

(قادر سازی فعال با سطح پایین active low enable)



E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

کد وریلاگ دیکدر دو-به-چهار

سطح گیت/ساختاری

```
module decoder_2x4_gates (D, A, B, enable);
  output      [0: 3]          D;
  input                       A, B;
  input                       enable;
  wire                        A_not,B_not, enable_not;

  not
    G1  (A_not, A),
    G2  (B_not, B),
    G3  (enable_not, enable);
  nand
    G4  (D[0], A_not, B_not, enable_not),
    G5  (D[1], A_not, B, enable_not),
    G6  (D[2], A, B_not, enable_not),
    G7  (D[3], A, B, enable_not);

endmodule
```

مدل سازی جریان دادگان Dataflow Modeling

این مدل سازی برای منطق ترکیبی شامل مجموعه عملگرهایی است که بر روی عملوندهای دودویی عمل می کند تا یک نتیجه دودویی حاصل شود.

این مدل سازی از تخصیص های پیوسته و کلمه کلیدی assign استفاده می کند.

این نوع مدل سازی HDL "کارکرد" را به جای ساختار گیت توصیف می کند.

یک کامپایلر سنتز قادر است ماژول توصیف شده به صورت جریان دادگان را به عنوان ورودی دریافت کرده و یک فهرست قطعات و یک طرح شماتیکی را فراهم کند.

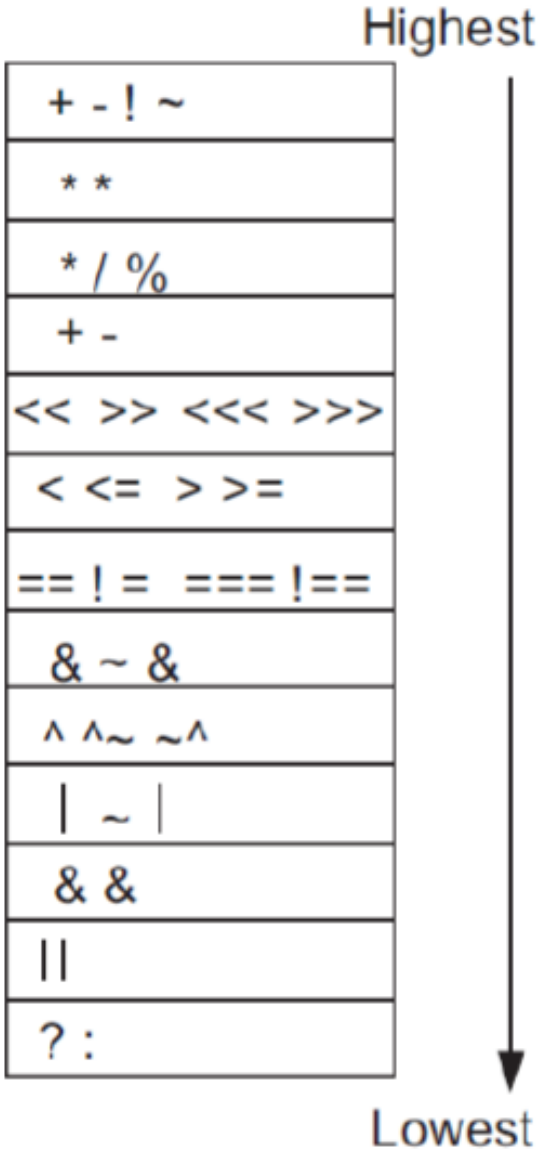
مدل سازی جریان دادگان

عملگرها

Operator type	Operator symbols	Operation performed	Number of operands
Bitwise	~	1's complement	1
	&	Bitwise AND	2
	|	Bitwise OR	2
	^	Bitwise XOR	2
	~^ or ^~	Bitwise XNOR	2
Logical	!	NOT	1
	&&	AND	2
	||	OR	2
Reduction	&	Reduction AND	1
	~&	Reduction NAND	1
	|	Reduction OR	1
	~|	Reduction NOR	1
	^	Reduction XOR	1
	~^ or ^~	Reduction XNOR	1
Arithmetic	+	Addition	2
	−	Subtraction	2
	−	2's complement	1
	*	Multiplication	2
	/	Division	2
Relational	>	Greater than	2
	<	Less than	2
	>=	Greater than or equal to	2
	<=	Less than or equal to	2
Equality	==	Logical equality	2
	!=	Logical inequality	2
Shift	>>	Right shift	2
	<<	Left shift	2
Concatenation	{,}	Concatenation	Any number
Replication	{{}}	Replication	Any number
Conditional	?:	Conditional	3

مدل سازی جریان دادگان

الویت عملگرها



مدل سازی جریان دادگان

کارکرد عملگرها

$$\sim(1010) \rightarrow (0101)$$

$$!(1010) \rightarrow 0$$

$$\textbf{if } ((A < B) \,\&\&\, (B < C)).$$

$$\{n\{1'b1\}\}$$

مدل سازی جریان دادگان

عملگر شرطی

`A ? B : C`

condition ? true-expression : false-expression;

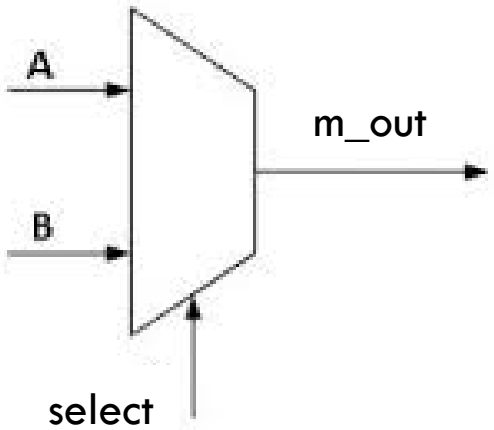
صفر منطقی معادل غلط

یک منطقی معادل درست

مثال:

```
module mux_2x1_df(m_out, A, B, select);
   output      m_out;
   input       A, B;
   input       select;

   assign m_out = (select)? A : B;
endmodule
```



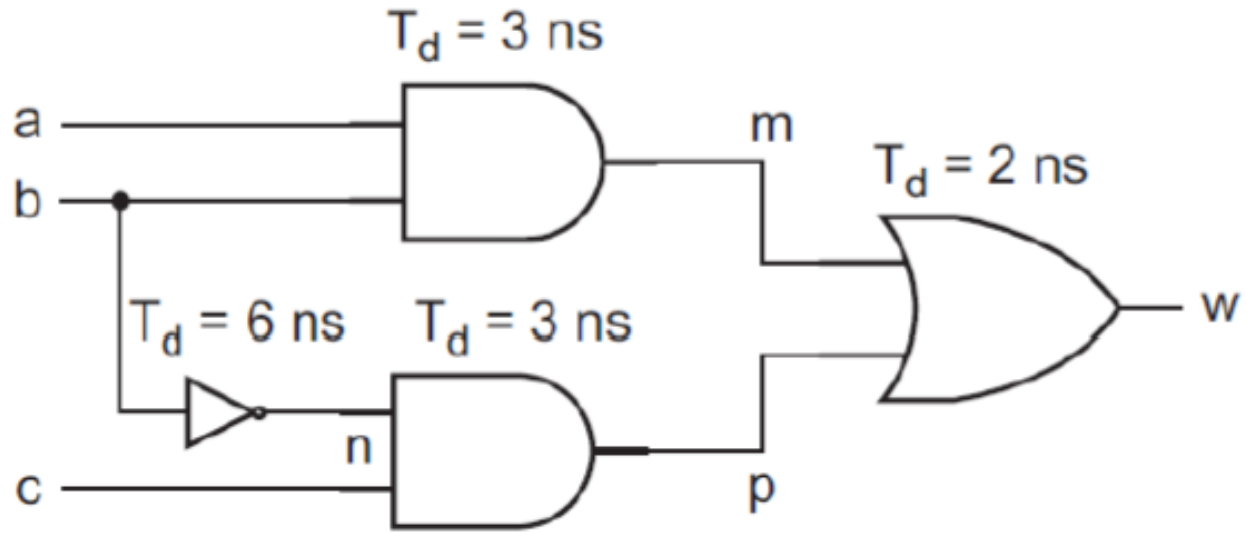
کد وریلاگ دیکدر دو-به-چهار

مدل سازی جریان دادگان

```
module decoder_2x4_df (
   output     [0: 3]          D,
   input                      A, B,
                              enable
);
   assign     D[0] = !((!A) && (!B) && (!enable)),
              D[1] = !( !A) && B && (!enable)),
              D[2] = !(A &&  (!B) && (!enable)),
              D[3] = !(A && B && (!enable));
endmodule
```

تاخیر در مدل سازی جریان دادگان

مثال:




```
assign #6 n = ~b;
assign #3 m = a & b;
assign #3 p = n & c;
assign #2 w2 = m | p;
```

تخصیص پیوسته

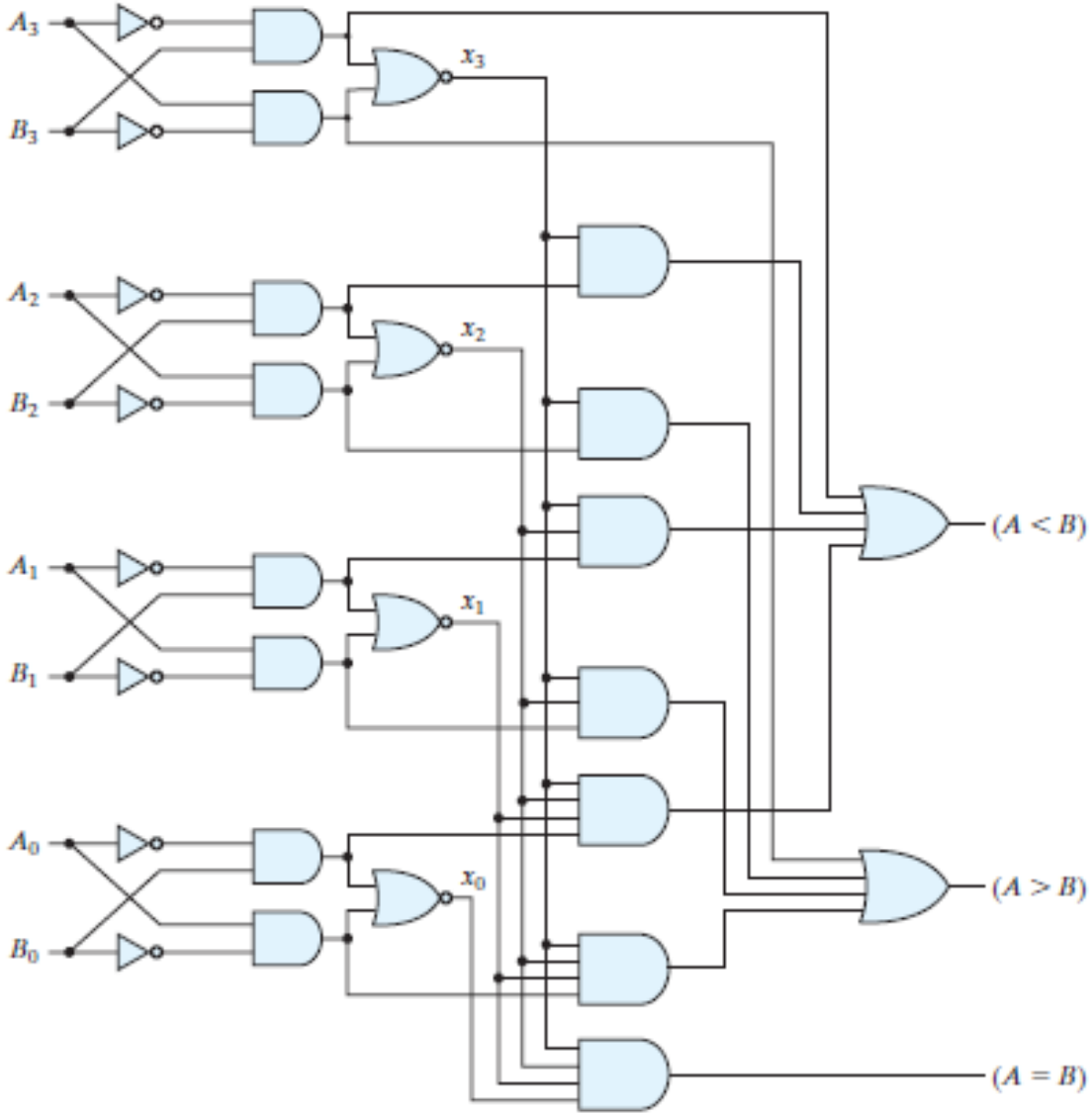
کلمه کلیدی assign تخصیص پیوسته را سبب می شود.

تخصیص های پیوسته ای که برای جریان دادگان تعریف می شود همیشه فعال هستند.

مادامی که سمت راست این نوع تخصیص تغییر کند این تغییر اعمال می شود ونتیجه به سمت چپ، تخصیص داده می شود.

مقایسه گر چهار بیتی

مقایسه A و B

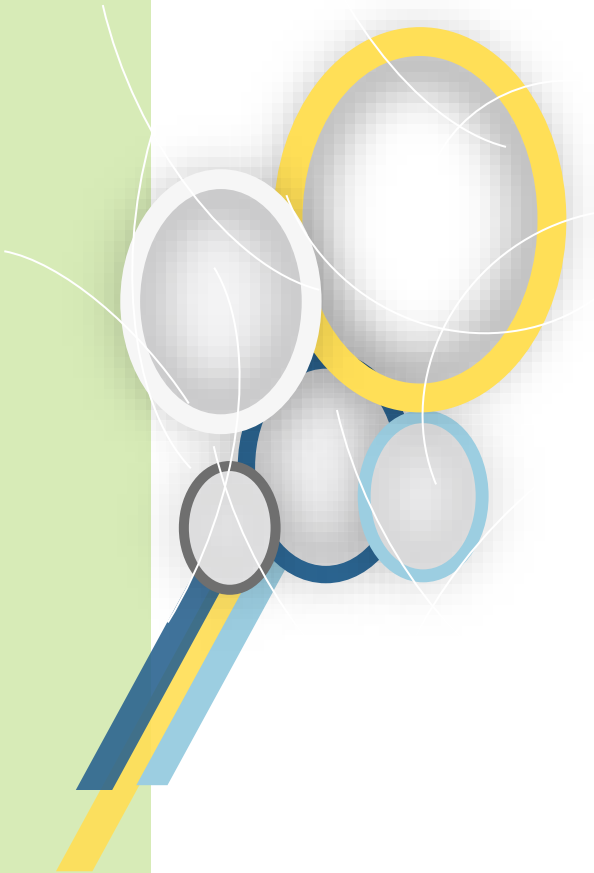


مقایسه گر چهار بیتی (ادامه)

مدل جریان دادگان

```
// Dataflow description of a four-bit comparator //V2001, 2005 syntax

module mag_compare
( output                A_lt_B, A_eq_B, A_gt_B,
  input [3: 0]          A, B
);
  assign A_lt_B = (A < B);
  assign A_gt_B = (A > B);
  assign A_eq_B = (A = = B);
endmodule
```



خسته نباشید