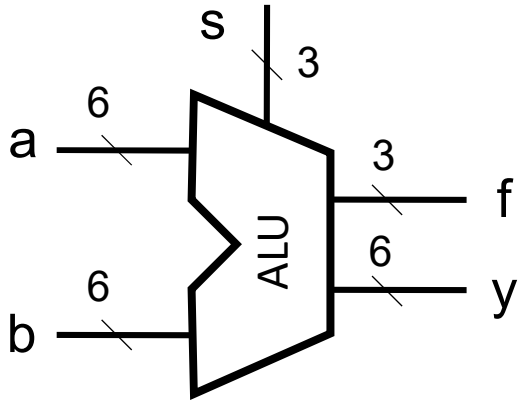
Lab1 Notes

zjx@ustc.edu.cn

2019.3.28

ALU 标志位

- 进位/借位（cf）
  - ADD：{ cf, y } = {1'b0, a} + {1'b0, b}
  - SUB：{ cf, y } = a - b
- 溢出（of）
  - ADD：of = (~a[5] & ~b[5] & y[5]) | (a[5] & b[5] & ~y[5])
  - SUB：of = (~a[5] & b[5] & y[5]) | (a[5] & ~b[5] & ~y[5])
- 零（zf）
  - zf = ~| y;



s	a[5]	b[5]	y[5]	of
+	0	0	1	1
	1	1	0	1
	其他			0
-	0	1	1	1
	1	0	0	1
	其他			0

溢出

- n位二进制补码表示范围：-2^{n-1} ~ $+2^{n-1}-1$
- 溢出：运算的结果超出了补码的表示范围
- **出现场合**
  - **同号相加，和的符号与被加数的符号相反**
  - **异号相减，差的符号与被减数的符号相反**
- 判别：最高位进位和次高位进位
  - 相同，则未溢出
  - 不相同，则溢出

参数化模块

- 传递子模块中定义参数的方法

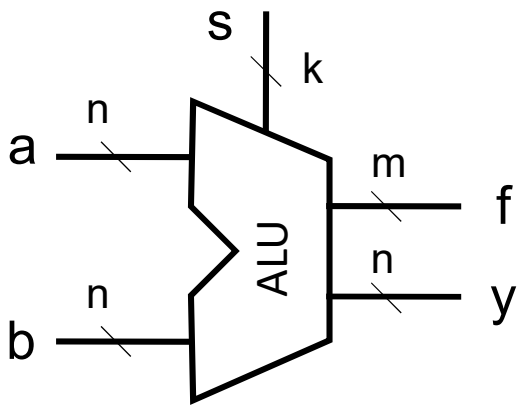
module_name #(parameter1, parameter2)
inst_name(port_map);

module_name
#(.parameter_name(para_value), .parameter_name(para_value)) inst_name (port map);

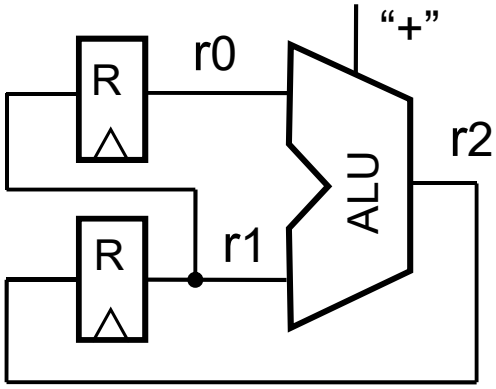
用#方法传递参数方法与port map的写法类似

参数化模块 — ALU

```
module  #(parameter MSB=31,
DELAY=2)  alu (
input [MSB : 0]  a, b,
……
);
parameter  ADD = 3'd0, SUB = 3'd1, …… ;
……  // 定义y, cf, of
assign #DELAY zf = ~|y;  //加参数仅用于示例
endmodule
```

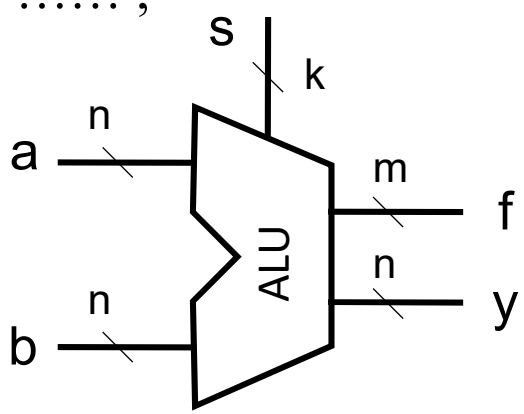



```
alu  ALU0 (r0, r1, …);
alu  #(8) ALU1 (.y(r2), .a(r0), …);
alu  #(.DELAY(5)) ALU2 (r0, r1, …);
```



参数化模块 — ALU

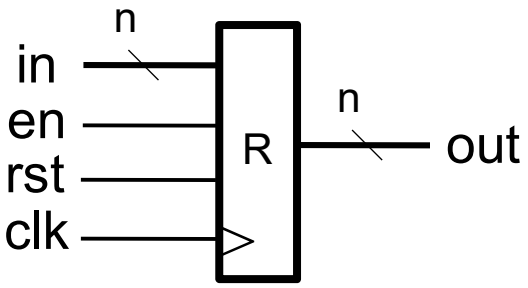
```
always  @ (*) begin    // 包含函数 (y, cf, of) 的所有自变量 (a, b, s)
      cf =0; of =0;       // 设置cf 和of的默认值，使得所有分支都有赋值
      case (s)
            ADD: begin
                  {cf, y} = a + b;
                  of = of = (~a[MSB] & ~b[MSB] & y[MSB]) | …… ;
            end
            SUB: begin
                  {cf, y} = a + b;
                  of = (~a[MSB] & b[MSB] & y[MSB]) | …… ;
            end
            AND: y = a & b;
            ……
            default: y = 0 ;  //设置y其他分支的取值
      endcase
end
```

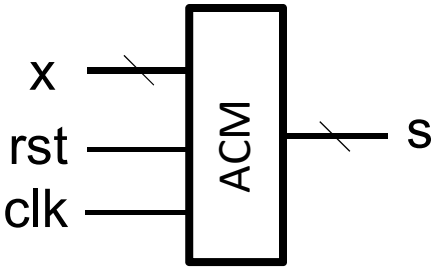


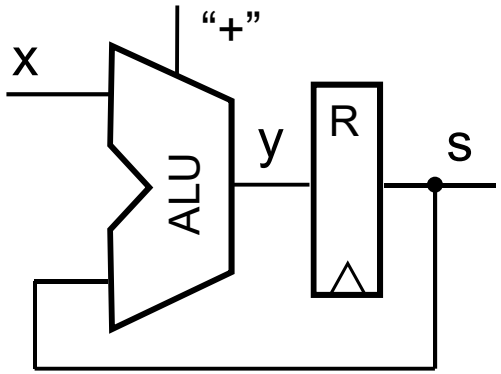
参数化模块 — 寄存器

```
module  # ( parameter N = 32,
RST_VALUE = 0 )  register (
input  [N - 1 : 0]  in,
input  en, rst, clk,
output reg  [N - 1 : 0] out
);
always @(posedge clk, posedge rst)
      if (rst) out <= RST_VALUE;
      else  if (en)
            out <= in;
endmodule

wire [15:0] y;
wire [15:0] s;
alu #(16) ALU (.a(x), .b(s), .s(ADD), .y(y));
register #(.N(16)) R (y, 1, rst, clk, s);
```

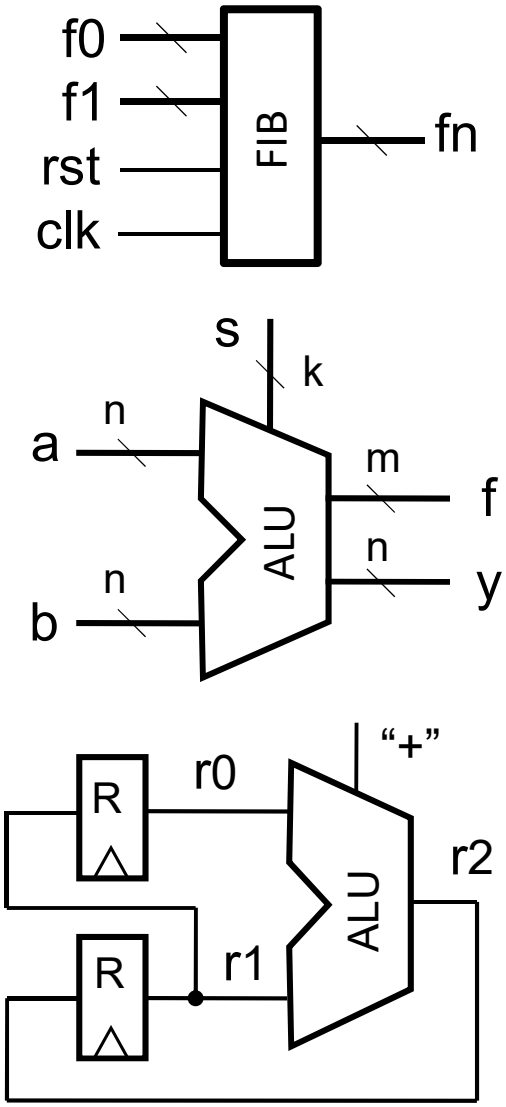






FIB

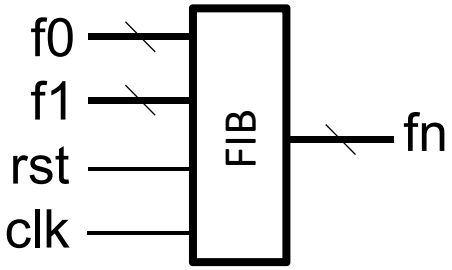
```
module  fib (
        input  [7 : 0]  f0, f1,
        input  rst, clk,
        output [7 : 0]  fn
);
localparam  ADD = 3'd0;
reg [7:0]  r0, r1;
wire [7:0]  r2;
alu  #(8) ALU (.y(r2), .a(r0), .b(r1), .s(ADD));
……  // r0 和r1逻辑
assign  fn = r0;      // or r1 or r2
endmodule
```

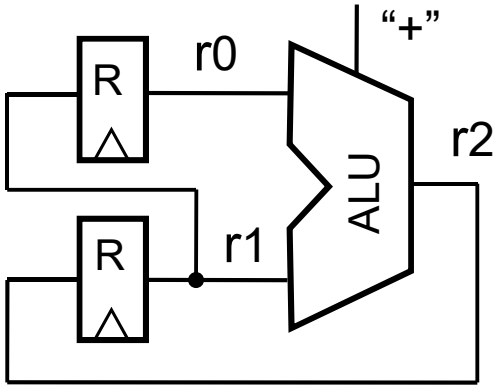


初始化寄存器

```
reg [7:0]  r0, r1;
// reg [7:0]  r0 = f0, r1= f1;
// initial  begin r0 = f0; r1=f1; end
……
always @(posedge clk, posedge rst) begin
      if (rst) begin
            r0 <= f0;
            r1 <= f1;
      end
      else  begin
            r0 <= r1;
            r1 <= r2;
      end
end
```

仅用于仿真，对于下载无用





手动时钟

- 用开关输入信号 (例如x) 作为时钟信号，必须在约束文件(.xdc)中设置

```
set_property  CLOCK_DEDICATED_ROUTE   FALSE
[get_nets  {x}]
```

The End