

kaikkecc Update README.md

Latest commit 5d25569 2 minutes ago

..

A11

m

4 hours ago

README.md

Update README.md

2 minutes ago

README.md



Atv_2715_011

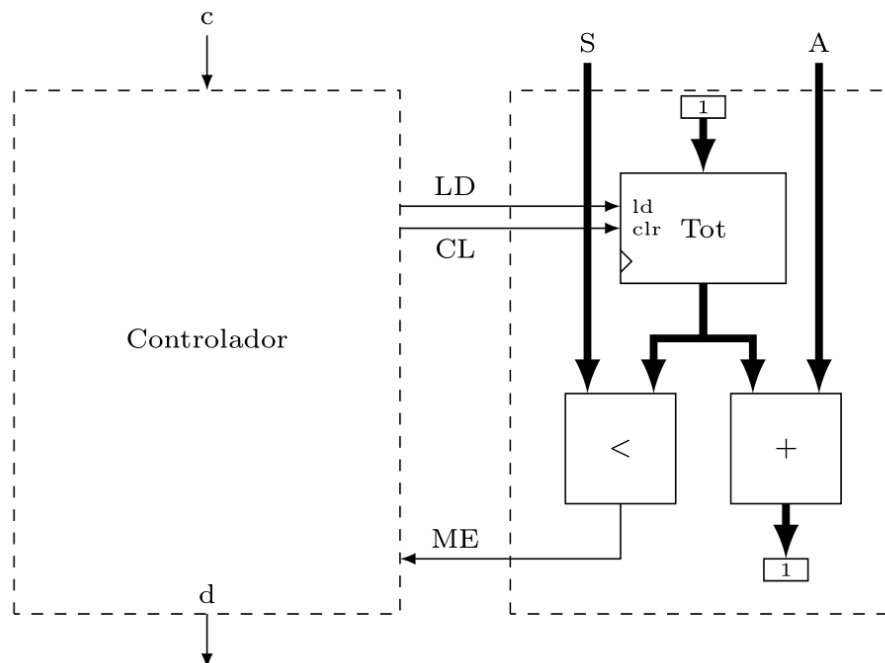
PROFESSOR Dr.: SAMAHERNI MORAIS DIAS

ESTUDANTE : KAIKE CASTRO CARVALHO

1. INTRODUÇÃO

1 - Projete um circuito lógico para uma máquina de vendas. A máquina possui duas entradas de dados, sendo uma para informar o valor da moeda inserida (A) e uma outra para entrar com o valor do produto (S). Também conta uma entrada para indicar quando uma moeda foi depositada ($c=1$) e uma saída ($d=1$) para liberar o produto.

Figura 1. Bloco Problema



2. OBJETIVO

Desenvolver um VHD que compara o valor de entrada com o preço do produto e libera ou não a compra.

3. DESENVOLVIMENTO

O projeto obedeceu o passo a passo em Nível de Transferência entre Registradores (RTL) que organizar a forma do bloco de controle se comunicar com o bloco operacional. As seções a seguir são as fases de extrair a solução de acordo com o enunciado do problema.

3.1 CRIAR UMA MÁQUINA DE ESTADOS DE ALTO NÍVEL

A máquina de estados foi desenvolvida pensando no processo que seria: início, esperar, somar e fornecer. As entradas c (bit), a (8 bits) e s (8 bits). As saídas d (bit), a (8 bits) e s (8bits). As saída é: d (bit).

Figura 2. Máquina de Estados

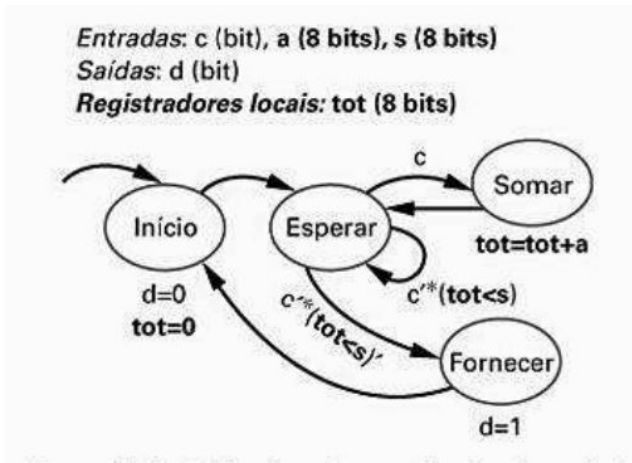


Tabela 1. Relação entre entradas e saídas

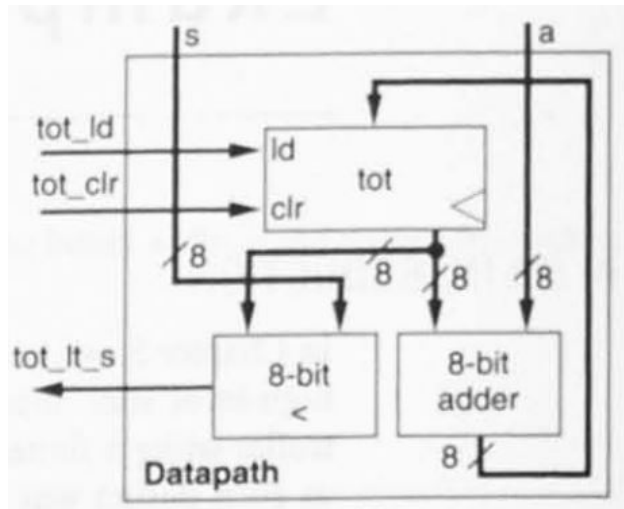
	ENTRADAS					SAÍDAS					
	s1	s0	c	tot_lt_s		d	tot_ld	tot_clr	n1	n0	
0	0	0	0	0		0	0	1	0	1	INICIAR
1	0	0	0	1		0	0	1	0	1	
2	0	0	1	0		0	0	1	0	1	
3	0	0	1	1		0	0	1	0	1	
4	0	1	0	0		0	0	0	1	1	ESPERAR
5	0	1	0	1		0	0	0	0	1	
6	0	1	1	0		0	0	0	1	0	
7	0	1	1	1		0	0	0	1	0	
8	1	0	0	0		0	1	0	0	1	SOMAR
9	1	0	0	1		0	1	0	0	1	
10	1	0	1	0		0	1	0	0	1	
11	1	0	1	1		0	1	0	0	1	
12	1	1	0	0		1	0	0	0	0	LIBERAR
13	1	1	0	1		1	0	0	0	0	
14	1	1	1	0		1	0	0	0	0	
15	1	1	1	1		1	0	0	0	0	

- Equações das saídas resultante da tabela 1:
- $d = s1*s0;$
- $tot_ld= s1' * s0';$
- $tot_clr= s1' * s0';$
- $n1= s1' * s0 * c*tot_lt_s' + s1' * s0 * c;$
- $n0= s0' + s1' * s0 * c';$

3.2 CRIAR UM BLOCO OPERACIONAL

As entradas *s* e *a* são de 8 bits e representam o valor do produto e a quantidade que será inserido na máquina de refrigerante, respectivamente. As outras entradas *tot_ld*, *tot_clr*, *tot_lt_s* são LD, LC e ME, respectivamente.

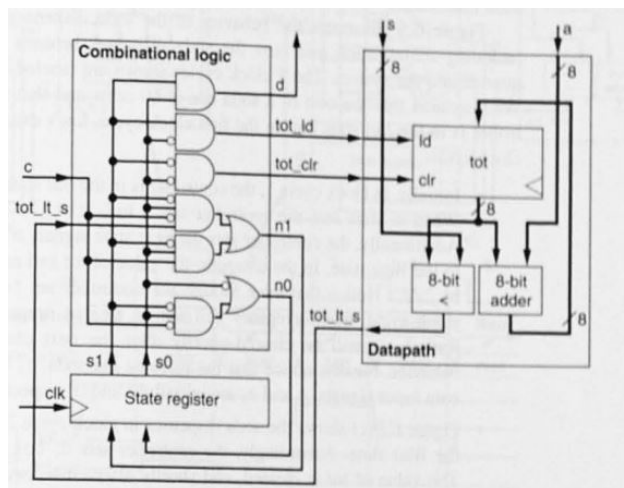
Figura 3. Bloco Operacional



3.3 CONECTAR O BLOCO OPERACIONAL AO BLOCO DE CONTROLE

A construção do bloco de controle possui um bloco lógico e um registrador de 2 bits que *c* representa quando uma moeda é inserida.

Figura 3. Visão completada dos blocos



4. RESULTADOS

- Entidade do projeto que expõem as entradas do circuito:

```
entity main is
    port(c, clock: in bit;
          S, A: in bit_vector(7 downto 0);
          d: out bit);
end;
```

- Arquitetura que relaciona os componentes datapath e controller seguindo o modelo RTL.

```

architecture circuito of main is

component datapath is

    port(aa,b: in bit_vector(7 downto 0);
          tot_ld, tot_clr, clk: in bit;
          tot_It_s: out bit);

end component;

component controller is

    port(cc, mme, clocck: in bit;
          LD, CL, dd: out bit);

end component;

signal ME: bit;
signal AUX: bit_vector(2 downto 0);

begin

CLLER: controller port map(

    cc => c,
    mme => ME,
    clocck => clock,

    LD => AUX(0),
    CL => AUX(1),
    dd => AUX(2));

DP: datapath port map(

    aa => A,
    b => S,
    tot_ld => AUX(0),
    tot_clr => AUX(1),
    clk => clock,
    tot_It_s => ME );
d <= AUX(2);

end circuito;

```

5. CONCLUSÃO

O circuito foi desenvolvido de acordo com o proposto pelo enunciado do problema.