Desenvolvimento de Sistemas Embarcados em Tempo Real

Prof. Hermano Cabral

Departmento de Eletrônica e Sistemas — UFPE

11 de setembro de 2017

Plano de Aula

Tema central

- Máquinas de estado

Plano de Aula

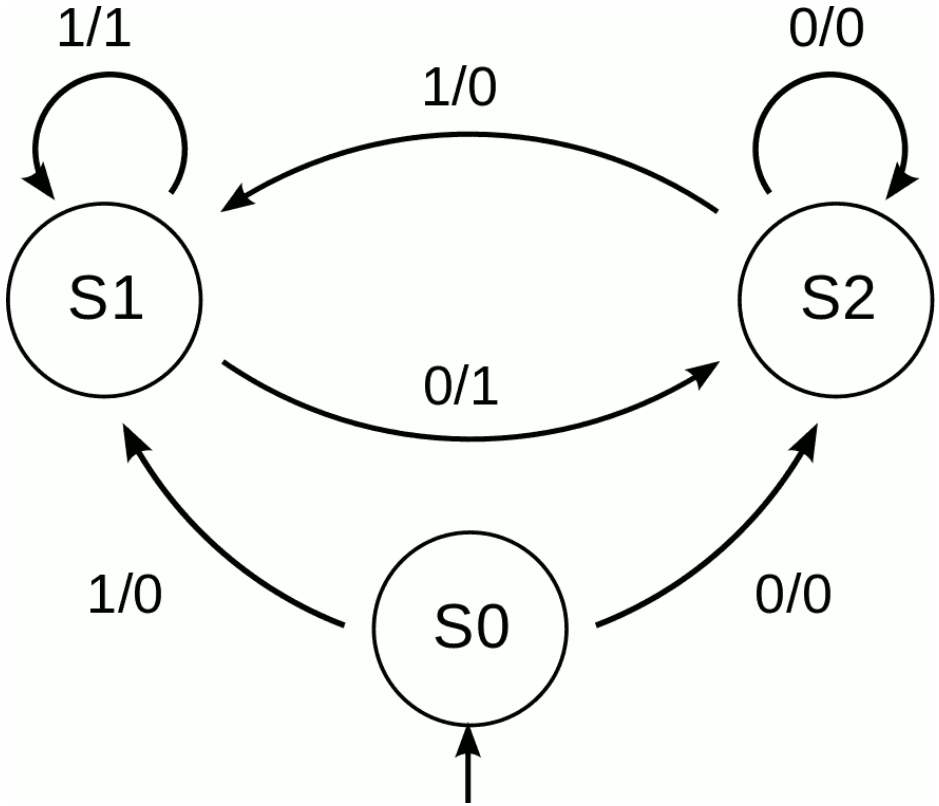
Tema central

- Máquinas de estado

Objetivos

- Conhecer as característica de uma máquina de estado
- Programar uma máquina de estados

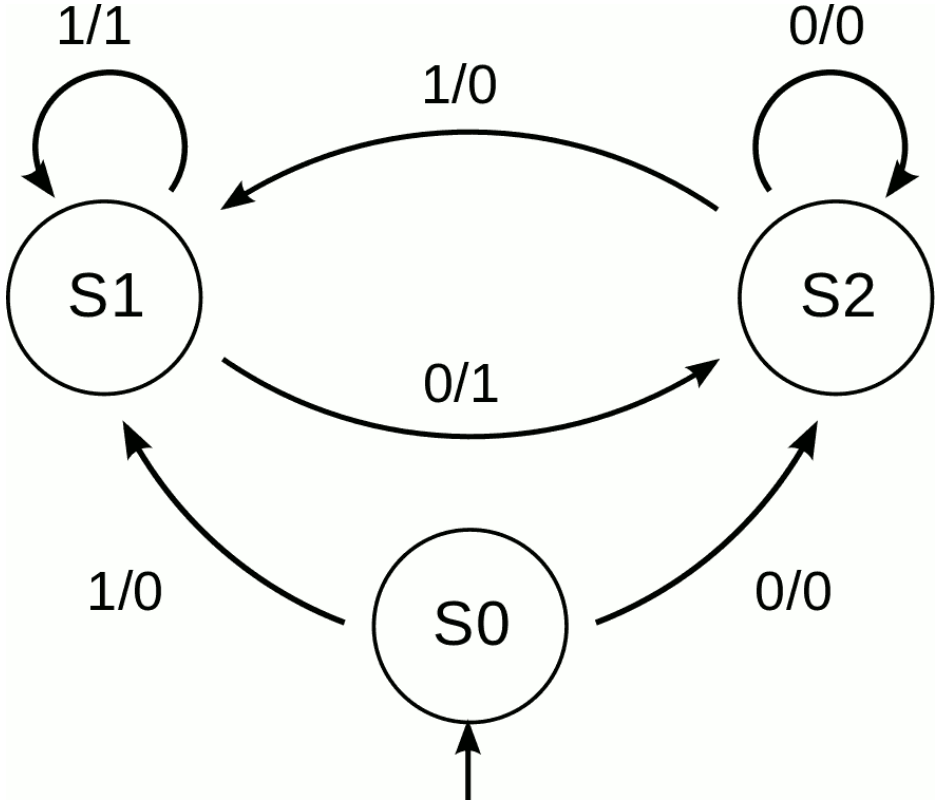
Máquina de estados



Definição

- Uma máquina de estados é um sistema com conjuntos finitos de estados, entradas e saídas e uma função de transição de estados.

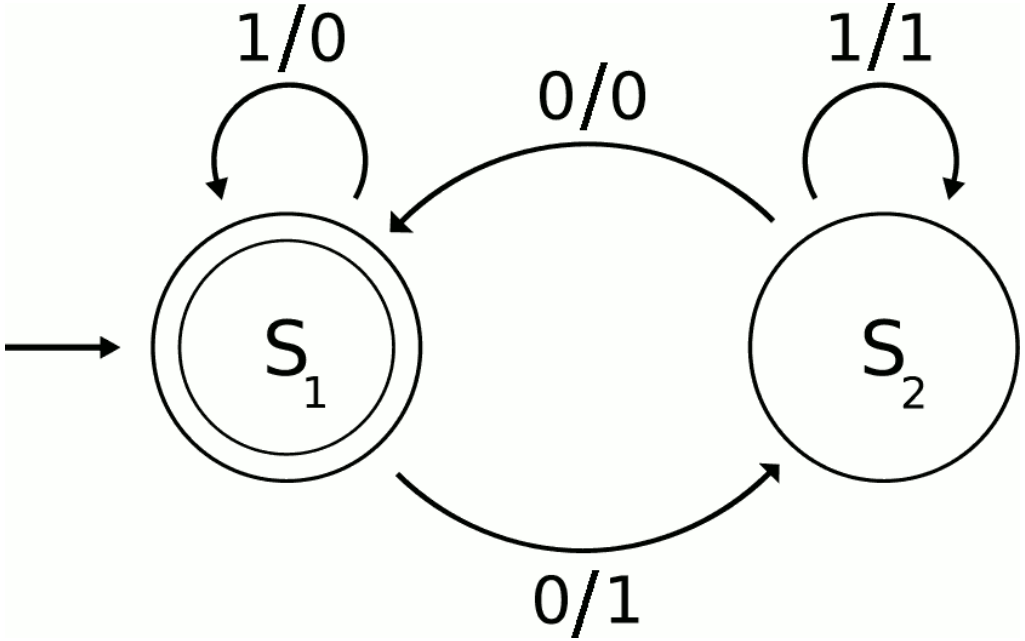
Máquina de estados



Definição

- Cada transição de um estado a outro é rotulado na forma e/s, onde e é a entrada do sistema e s é a saída.

Máquina de estados

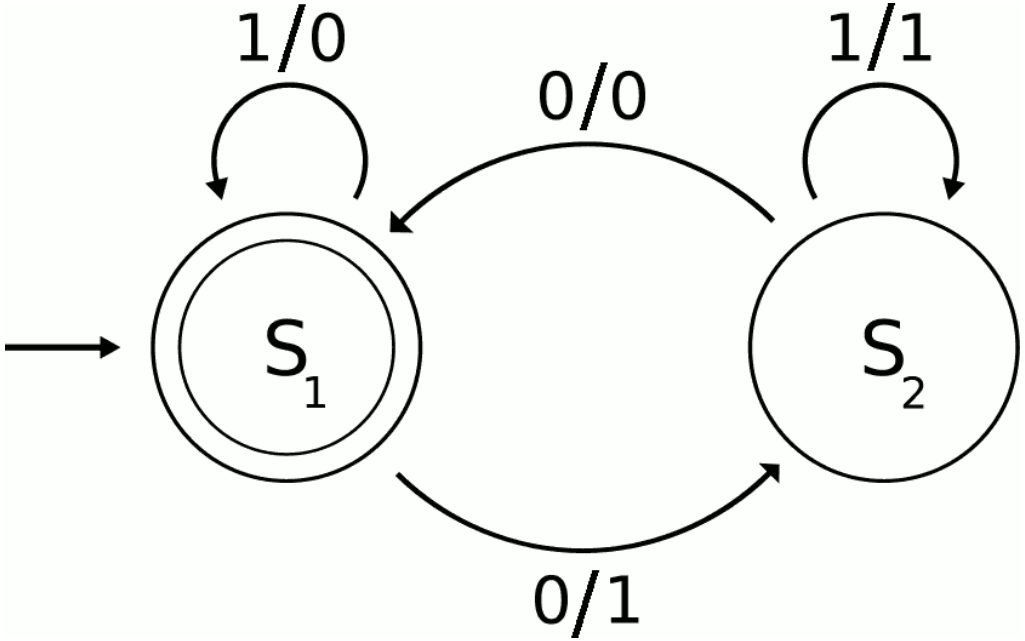


Máquina de estados para determinar se uma sequência de bits possui um número par ou ímpar de 0

Características

- Uma máquina de estados é útil na representação de um sistema onde um estado pode representar todo o histórico de eventos ocorridos do sistema.

Máquina de estados

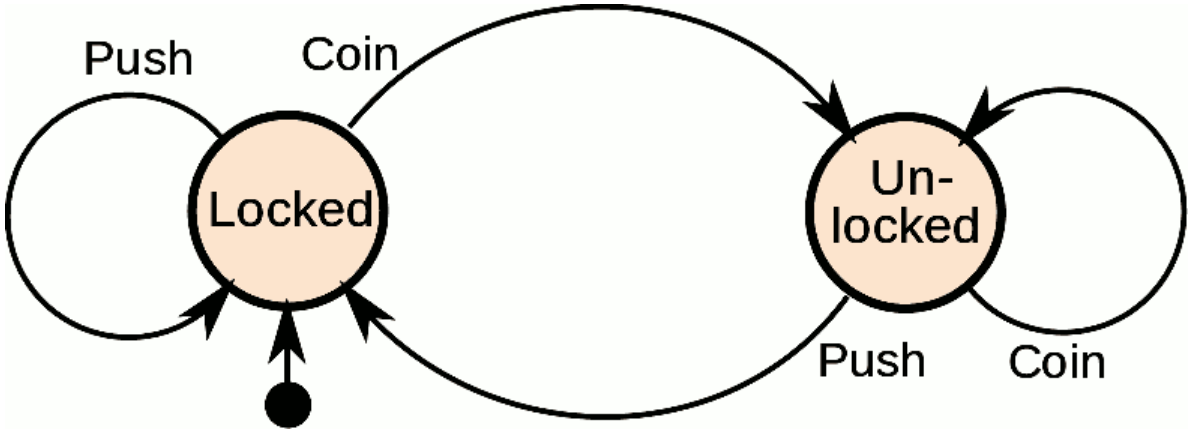


Máquina de estados para determinar se uma sequência de bits possui um número par ou ímpar de 0

Características

- Uma máquina de estados é útil na representação de um sistema onde um estado pode representar todo o histórico de eventos ocorridos do sistema.
- Isto significa que a resposta do sistema a um evento depende

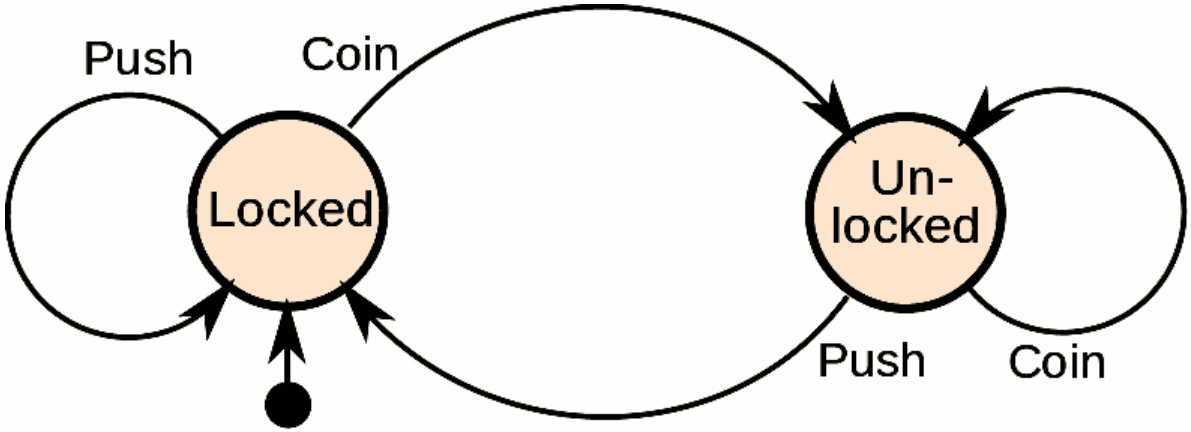
Máquina de estados



Características

- Um outro exemplo é o de uma catraca de metrô.

Máquina de estados



Características

- Um outro exemplo é o de uma catraca de metrô.
- Observe que no exemplo acima os eventos são {moeda inserida, catraca empurrada}, e não números.

Máquina de estados

Representação

- Além da representação visual, uma máquina de estados também pode ser representada por uma tabela.

Representação

- Além da representação visual, uma máquina de estados também pode ser representada por uma tabela.
- No caso do número par ou ímpar de zeros:

	S_1	S_2
0	$S_2/1$	$S_1/0$
1	$S_1/0$	$S_2/1$

Máquina de estados

Implementação

- Uma máquina d eestados pode ser implementada de diferentes maneiras em C.

Máquina de estados

Implementação

- Uma máquina d eestados pode ser implementada de diferentes maneiras em C.
- Duas formas mais usuais são:
  - Usando a instrução `switch`
  - Usando um vetor de ponteiros para funções

Máquina de estados

```
 1 typedef enum {
 2     STATE1 = 0, STATE2
 3
 4 } estados;
 5 typedef enum {
 6     INPUT1 = 0, INPUT2
 7 } entradas;
 8
 9 int main()
10 {
11     estados state  = STATE1;
12     entradas input = INPUT1;
13
14     switch (state) {
15         case STATE1:
16             switch (input) {
17                 case INPUT1:
18                     break;
19                 case INPUT2:
20                     break;
21                 default:
22                     break;
23             }
24             break;
25         case STATE2:
26             switch (input) {
27                 case INPUT1:
28                     break;
29                 case INPUT2:
30                     break;
31                 default:
32                     break;
33             }
34             break;
35         default:
36             break;
37     }
38
39     return 0;
40 }
```

implementação – usando switch

- A implementação por um switch duplo está mostrada ao lado.

Máquina de estados

```
 1 typedef enum {
 2     STATE1 = 0, STATE2
 3
 4 } estados;
 5 typedef enum {
 6     INPUT1 = 0, INPUT2
 7 } entradas;
 8
 9 int main()
10 {
11     estados state  = STATE1;
12     entradas input = INPUT1;
13
14     switch (state) {
15         case STATE1:
16             switch (input) {
17                 case INPUT1:
18                     break;
19                 case INPUT2:
20                     break;
21                 default:
22                     break;
23             }
24             break;
25         case STATE2:
26             switch (input) {
27                 case INPUT1:
28                     break;
29                 case INPUT2:
30                     break;
31                 default:
32                     break;
33             }
34             break;
35         default:
36             break;
37     }
38
39     return 0;
40 }
```

implementação – usando switch

- A implementação por um switch duplo está mostrada ao lado.
- Note que podemos trocar a instrução `switch` por uma sequência de `if`s.

Máquina de estados

```
 1 typedef enum {
 2     STATE1 = 0, STATE2, MAX_STATE
 3 } estados;
 4 typedef enum {
 5     INPUT1 = 0, INPUT2, MAX_INPUT
 6 } entradas;
 7 typedef void (*cb_t)(void);
 8
 9 cb_t machine[MAX_STATE][MAX_INPUT] = {
10     0, 0,  /* funções para estado 1 */
11     0, 0   /* funções para estado 2 */
12 };
13 estados next_state[MAX_STATE][MAX_INPUT] = {
14     0, 0,  /* próximos estados para estado 1 */
15     0, 0   /* próximos estados para estado 2 */
16 };
17 estados state  = STATE1;
18
19 int main()
20 {
21     entradas input;
22
23     while(1) {
24         input = wait_for_events();
25         machine[state][input];
26         state = next_state[estado];
27     }
28
29     return 0;
30 }
```

Implementação — usando ponteiros

- Ao invés de um switch duplo, podemos usar uma tabela de ponteiros para funções.

Máquina de estados

Exemplo

- Implemente um programa que retire as linhas de comentário de um programa C.