

Implementação de Máquinas de Estados

Prof. Hugo Vieira Neto

2020/2

Exercício 1 (Teórico)

- Esboce um diagrama de estados que modele o comportamento dinâmico das trocas de modo de operação de um núcleo ARM Cortex-M4
- Considere apenas a situação de operação em que os registradores especiais PRIMASK = 1, FAULTMASK = 1 e CONTROL = 0
- Comece questionando quais são os eventos de interesse para a situação de operação acima

Exercício 1 (Teórico)

- Considere estados intermediários de *stacking*, *unstacking* e *tail-chaining*, se for o caso
- Na representação das transições ocasionadas pela execução de uma instrução que ocasione retorno de exceção (ex: BX LR), considere possíveis exceções pendentes (bit PEND)
- Utilize a notação UML para a caracterização das transições: evento[condição]/ação

Seleção por Estado

```
typedef enum {Estado_0, Estado_1, Estado_2} state_t;
volatile uint8_t Evento = 0; // alterado por ISR

void tarefa(void) {
    state_t Estado = Estado_0; // estado inicial da MEF
    while(1) {
        switch(Estado) {
            case Estado_0:
                if(Evento == 1) {
                    // ações e mudança de estado
                } // if
                break;
                ...
            } // switch
        } // while
    } // tarefa
```

Projeto “fsm_state” da área de
trabalho “EK-TM4C1294XL_IAR8”

Exercício 2 (Seleção por Estado)

- Esboce um diagrama de estados que descreva o comportamento dinâmico do projeto “fsm_states”
- Como implementar ações de entrada e saída (entry / e exit /) nos estados e suas atividades (do /) na abordagem de seleção por estado?

Seleção por Evento

```
typedef enum {Estado_0, Estado_1, Estado_3} state_t;
volatile uint8_t Evento = 0; // alterado por ISR

void tarefa(void){
    state_t Estado = Estado_0; // estado inicial da MEF
    while(1){
        if(Evento){
            switch(Estado){
                case Estado_0:
                    // ações e mudança de estado
                    break;
                :
            } // switch
        } // if
    } // while
} // tarefa
```

Projeto “fsm_event” da área de
trabalho “EK-TM4C1294XL_IAR8”

Exercício 3 (Seleção por Evento)

- Altere o projeto “fsm_event” de forma a apresentar a sequência crescente do Código de Gray de 3 bits nos LED D1, D2 e D3 do kit EK-TM4C1294XL
- Sugestão: utilize um estado diferente para cada padrão binário de saída
 - 000 → 001 → 011 → 010 → 110 → 111 → 101 → 100 → 000 → ...

Implementação Matriz Estado-Evento

```
state_t f_0(state_t curr){
    // ações
    return next; // mudança de estado
} // f_0
⋮
void tarefa(void){
    state_t (*matriz[N][M])(state_t) = {{f_0, f_1, f_2},
                                          {f_3, f_4, f_5}};

    state_t Estado = Estado_0; // estado inicial da MEF
    while(1){
        if(Evento){
            Estado = (*matriz[Evento - 1][Estado])(Estado);
            Evento = 0;
        } // if
    } // while
} // tarefa
```

Projeto “fsm_matrix” da área de
trabalho “EK-TM4C1294XL_IAR8”

Exercício 4 (Seleção por Matriz)

- Esboce um diagrama de estados que utilize o conceito de hierarquia para descrever o comportamento dinâmico do projeto “fsm_matrix”
- Como implementar ações de entrada e saída (entry / e exit /) nos estados e suas atividades (do /) na abordagem de seleção por matriz estado-evento?

Ferramentas Úteis

- Diagrams.net (desenho na nuvem)
 - <https://app.diagrams.net/>
- UMLetino (desenho na nuvem)
 - <http://www.umlet.com/umletino/umletino.html>
- Yakindu Statechart Tools (simulação)
 - <http://statecharts.org/>