
1: Periféricos

- (1) Liste a funcionalidade dos periféricos listados a seguir : • RTC - Real time clock /
• TC - Timer/Counter

Segundo o manual do SAME70, o Real Time clock é o periférico responsável por manter controle do tempo presente, ou seja, como o nome diz, um circuito integrado que simula um relógio no microcontrolador. Sua principal função é conseguir dizer o horário atual quando requisitado.

Ja o timer/counter é um periférico especializado em contar intervalos de tempo, ou seja, usado para verificar o tempo de execução de algum evento.

- (2) Qual endereço de memória reservado para os periféricos ? Qual o tamanho (em endereço) dessa secção ?

O endereço de memória reservado para os periféricos, como pode ser visto na figura da página 41 do manual do Cortex M7, é o 0x40000000. O tamanho dessa seção é de 30bits.

- (3) Encontre os endereços de memória referentes aos seguintes periféricos : 1. PIOA
2. PIOB 3. ACC 4. UART1 5. UART2

Como é listado no manual do Cortex M7, os endereços de memória são os seguintes:

PIOA = 0x400E0E00

PIOB = 0x400E1000

ACC = 0x40044000

UART1 = 0x400E0A00

UART2 = 0x400E1A00

2: PMC - Gerenciador de energia

- (1) Qual ID do PIOC ?

O id do PIOC, segundo o manual do Cortex M7, é 12.

3: Parallel Input Output (PIO)

- (1) Verifique quais periféricos podem ser configuráveis nos I/Os : 1. PC1 2. PB6

Para o I/O PC1, segundo o manual do Cortex M7, podem ser configuráveis os periféricos D1 e PWMC0 PWML1

Já no I/O PB6, não há nenhum periférico configurável.

- (2) O que é debouncing ? Descreva um algoritmo que implemente o debouncing.

Como pode ser visto em [1], debouncing seria um tipo de hardware ou software que assegura um único sinal digital quando um contato for aberto ou fechado, isso é necessário pois há um fenômeno chamado bouncing, no contato de dois metais em um dispositivo eletrônico, há uma tendência que sejam gerados múltiplos sinais digitais em vez de um só.

(3) O que é race conditions ? Como que essa forma de configurar os registradores evita isso?

Segundo [2], race conditions são situações em que um computador ou sistema eletrônico tenta ler ou alterar uma quantidade de dados recebidas quase ao mesmo tempo, assim o sistema tenta sobrescrever os dados enquanto dados antigos ainda estão sendo lidos. Esse fenômeno pode resultar em um travamento do sistema, "operação ilegal", encerramento do sistema, erros de leitura dos dados antigos ou erros na alteração dos novos dados.

Com essa configuração de registradores, não há conflitos de leitura ou alteração de dados por um mesmo registrador, cada registrador fica responsável por executar uma tarefa específica, por exemplo: PIO_CODR é responsável apenas pelo clear do pino e PIO_SODR é responsável apenas pela gravação do pino.

(4) Explique com suas palavras o trecho anterior extraído do datasheet do uC, se possível referencie com o diagrama "I/O Line Control Logic".

Esse trecho explica a função de alguns registradores na configurações de pinos. Primeiramente podemos atribuir um valor nos registradores PIO_ABCDSR1 e PIO_ABCDSR2 para determinar se o controlador PIO irá energizar o pino. Após isso, pode-se configurar se um pino será de entrada ou saída com os registradores PIO_OER e PIO_ODR. Por fim, deve-se usar o registrador PIO_CODR para gravar um valor no pino ou PIO_SODR para zerar esse valor, assim determinado o valor de saída do pino.

Referências

- [1] *Debouncing*. Disponível em <http://whatis.techtarget.com/definition/debouncing>, Acessado em 22-02-2017.
- [2] *RACE CONDITION*. Disponível em <https://www.vivaolinux.com.br/artigo/Race-Condition>, Acessado em 22-02-2017.