

Pesquisa 07 – PIO Output

Pedro Cunial

February 22, 2017

Contents

1	Periféricos	1
1.1	Liste a funcionalidade dos periféricos listados a seguir:	1
1.2	Mapa de memória	2
1.3	Encontre os endereços de memória referentes aos seguintes periféricos:	2
2	PMC - Gerenciador de energia	2
2.1	Qual o ID do PIO?	2
3	Parallel Input Output (PIO)	3
3.1	Verifique quais periféricos podem ser configuráveis nos I/Os .	3
3.2	Debouncing	3
3.3	Race conditions	3
3.4	Explique com suas palavras o trecho anterior extraído do datasheet do uC, se possível referencie com um diagrama "I/O Line Control Logic".	4

1 Periféricos

1.1 Liste a funcionalidade dos periféricos listados a seguir:

- RTC – Real Time Clock

O RTC, assim como seu nome sugere, é um periférico capaz de manter informações sobre tempo atual (segundo, hora, dia etc) e fornecê-las. O RTC possui uma memória SRAM, normalmente de 56 bytes e é capaz de notificar um possível uP central de falhas em seu funcionamento. A comunicação do RTC é realizada pelo protocolo I2C.

- TC – Timer/Counter

Em contrapartida ao RTC, o Timer não guarda informações sobre a data precisa atual, mas sim lida com uma contagem de um dado tempo e é normalmente utilizado para gerar intervalos específicos marcados por tempo (um exemplo análogo ao componente na vida real seria uma ampulheta). Já o Counter, por sua vez, guarda a quantidade de vezes que um dado evento ocorrera dentro de um dado intervalo de tempo, como rotações por minuto de um motor, requisições por minuto à um componente etc.

1.2 Mapa de memória

- Qual endereço de memória reservado para os periféricos?
Os endereços de memória de periféricos vão de 0x40000000 até 0x60000000.
- Qual o tamanho (em endereço) dessa secção?
A secção possui 0x20000000 endereços possíveis.

1.3 Encontre os endereços de memória referentes aos seguintes periféricos:

- PIOA
0x400E0E00
- PIOB
0x400E1000
- ACC
0x40044000
- UART1
0x400E0A00
- UART2
0x400E1A00

2 PMC - Gerenciador de energia

2.1 Qual o ID do PIOC?

O ID do PIOC é 12

3 Parallel Input Output (PIO)

3.1 Verifique quais periféricos podem ser configuráveis nos I/Os

- PC1

O PC1 pode ser configurado como D1 ou PWMC0PWML1.

- PB6

O PB6 funciona exclusivamente como SWDIO/TM5.

3.2 Debouncing

- O que é debouncing?

O debouncing é uma técnica de garantir programaticamente que um botão tenha sido apertado somente uma vez. A necessidade do debouncing surge do fato que ao apertar um botão um sinal muito ruidoso é gerado, podendo, as vezes, gerar um sinal análogo à um duplo clique, ou seja, filtrar o sinal de clique de um botão para garantir que será computado apenas o verdadeiro clique no mesmo.

- Descreva um algoritmo que implemente o debouncing.

Um algoritmo que implemente o debouncing poderia seguir assim: Ao apertar o botão, segure sua thread por um pequeno intervalo, se após este pequeno intervalo o botão continuar apertado, isso significa que sua leitura de fato fora de um único clique, caso contrário, sua leitura foi, provavelmente, de algum ruído e deve ser ignorada.

3.3 Race conditions

- O que são race conditions?

Race conditions são falhas de concorrências em um sistema computacional, normalmente atrelada a não expectativa da necessidade de um dado processo em outro.

- Como que essa forma de configurar registradores evita isso?

Ao permitir a mudança de dados de um registrador somente por um operador externo, evita-se que os dados do mesmo sejam alterados como consequência em alguma outra operação do programa, tornando menos race conditions cada vez menos prováveis.

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

Para estabelecer a comunicação entre um periférico e um uC é necessário, primeiramente, definir qual dos dois será o controlador da relação, ou seja, qual dos dois "enviará" dados para o outro. No SAM-E70, podemos definir um periférico como controlador de uma comunicação ao "settar" seu respectivo pino de comunicação em 0. Por exemplo: Ao definir o PIO_{PSR} como 0, seus respectivos periféricos serão os ativos na comunicação com o uC. Da mesma forma, ao definir o valor do PIO_{PSR} como 1, definimos a comunicação como guiada pelo controlador. Neste caso, o periférico escreve seu valor de saída mais recente em um registrador de output exclusivo para o mesmo (PIO_{CODR}).