

Computação Embarcada-Pesquisa PIO Output

Felipe Frid Buniac

February 22, 2017

1 Periféricos

1.1 Periféricos

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

RTC:

Um "real time clock" (RTC) é um relógio de computador (na maioria das vezes na forma de um circuito integrado) que mantém o controle do tempo atual[1].

TC:

Um "counter" é geralmente construído de diversos flip-flops conectados em cascata. Os contadores são um componente amplamente utilizado em circuitos digitais e são fabricados como circuitos integrados separados e também incorporados como partes de circuitos integrados maiores [2].

1.2 Mapa de memória

Qual endereço de memória reservado para os periféricos ?

0x40000000[3]

qual o tamanho (em endereço) dessa secção ?

Possui 0x20000000 endereços possíveis[3]

1.3 Periféricos

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

Os endereços de memórias são:[3]

1. PIOA - 0x400E0E00
2. PIOB - 0x400E1000
3. ACC - 0x40044000
4. UART1 - 0x400E0A00
5. UART2 - 0x400E1A00

2 PMC - Gerenciador de energia

2.1 PIO D ID

Qual ID do PIOC ?

O ID do PIOC é 12.[3]

3 Parallel Input Output (PIO)

3.1 PIO periféricos

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

PC1 = Do tipo A o periférico D1 e do tipo B o periférico PWMC0_PWML1[3]

PB6 = Nenhum periférico de tipo A nem B[3]

3.2 Debouncing

O que é debouncing ? Descreva um algoritmo que implemente o debouncing.

Bouncing é a tendência de quaisquer dois metais terem contato com um dispositivo eletrônico para gerar sinais múltiplos como os contatos de fechar ou abrir. Debouncing é qualquer tipo de dispositivo de hardware ou software que garante que apenas um único sinal será ativo para uma única abertura ou fechamento de um contato. Um exemplo é o de, ao pressionar uma tecla no teclado do computador, você espera que um único contato seja gravado pelo computador [4].

Exemplo:

1. Initially set up a shift register variable to xFF
2. Using a timer set up a sampling event with a period (say 1 ms)
3. On the sample event:
4. Shift the variable towards MSB
5. Set LSB to current switch value
6. **if** shift register value =0 **then**
7. Set internal switch state to pressed
8. **else**
9. Set internal switch state to released
0. **end if**

Figure 1: Algoritmo de Bouncing [5]

3.3 Race conditions

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

Uma "race condition" ocorre quando dois "threads" acessam uma variável compartilhada ao mesmo tempo. O primeiro thread lê a variável e o segundo thread lê o mesmo valor da variável. Em seguida, o primeiro thread e o segundo thread executam suas operações no valor e "correm" para ver qual thread pode escrever o valor passado para a variável compartilhada. O valor do thread que escreve seu último valor é preservado, porque o segmento está escrevendo sobre o valor que o thread anterior escreveu.[6] Para evitar isto deveria ser feito algo com a memória para que, quando dois comandos, um de leitura e um de escrita são chamados ao mesmo tempo o computador sempre executa o de leitura antes de começar o comando de escrita. Desta forma não haveria o problema de "race condition".

3.4 Pino em modo output

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

Uma I/O line pode funcionar de duas maneiras diferentes. Esta pode ser controlada por um periférico, quando utilizada por uma função periférica, neste caso o bit assinalado é de zero com o pino podendo ser driven ou não. Outra forma de funcionamento é ser controlada por um PIO Controller. Neste caso o output pode estar habilitado ou desabilitado. Quando este drive é controlado pelo PIO Controller o bit registrado como zero é utilizado como apenas input e o bit utilizado como um, a linha é controlada pelo PIO Controller. Estes resultados podem ser vistos no PIO_ODSR.

References

- [1] https://en.wikipedia.org/wiki/Real-time_clock
- [2] [https://en.wikipedia.org/wiki/Counter_\(digital\)](https://en.wikipedia.org/wiki/Counter_(digital))
- [3] Cortex-M7-SAM-E70.pdf (datasheet)
- [4] <https://www.quora.com/What-is-debouncing>
- [5] <https://electrosome.com/switch-debouncing/>
- [6] <https://support.microsoft.com/en-us/help/317723/description-of-race-conditions-and-deadlocks>

4 GITHUB

<https://github.com/febuniac/EmbeddedComputing>