Aula 01 - Ricardo

Revisão

1A) O que é um sistema de Repositório de sotfware, e como ele é utilizado (git,svn mercurium) ?

Resposta:
Utilizado para Armazenar, gerir e documentar diferentes versões de software, linguagem ou todo o sistema operacional.
repositório de software é um local de armazenamento de onde pacotes de *software* podem ser recuperados e instalados em um computador

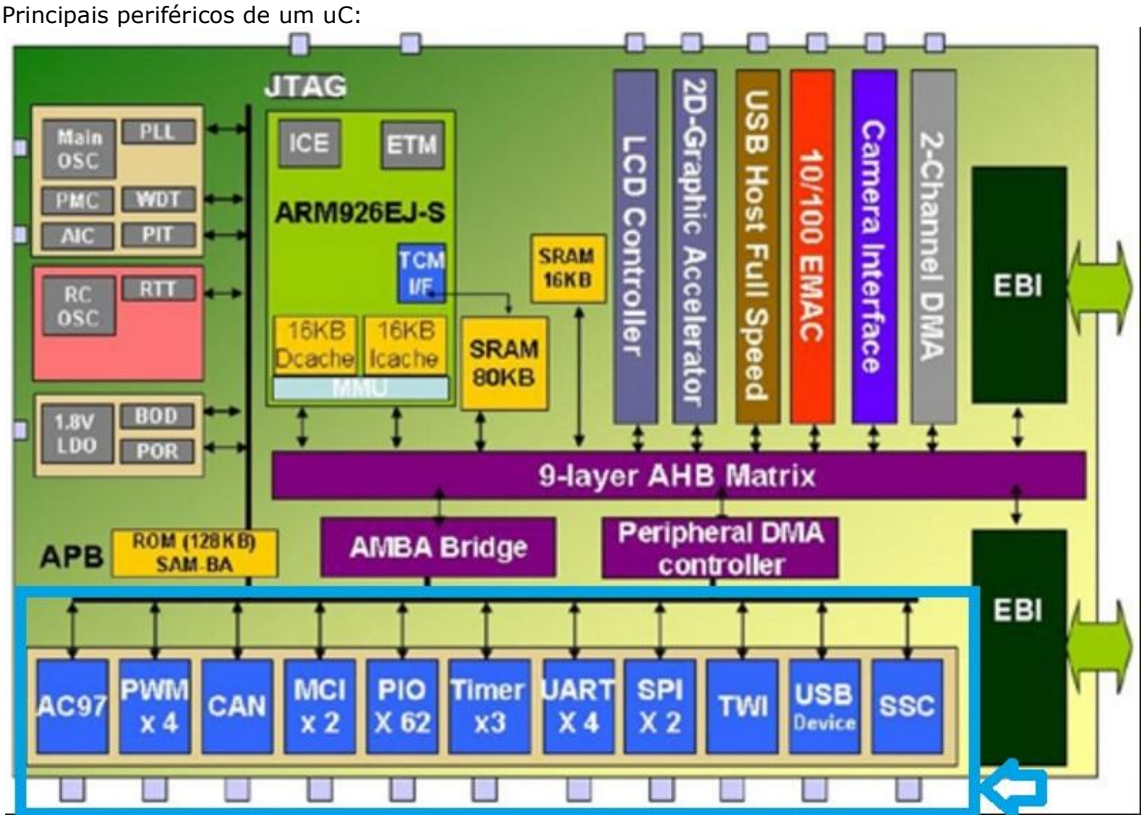
---

1B) Quais os principais fabricantes de Microcontroladores?

Resposta:
Principais fabricantes Microcontroladores
Resp; Atmel, Microchip, Texas instruments

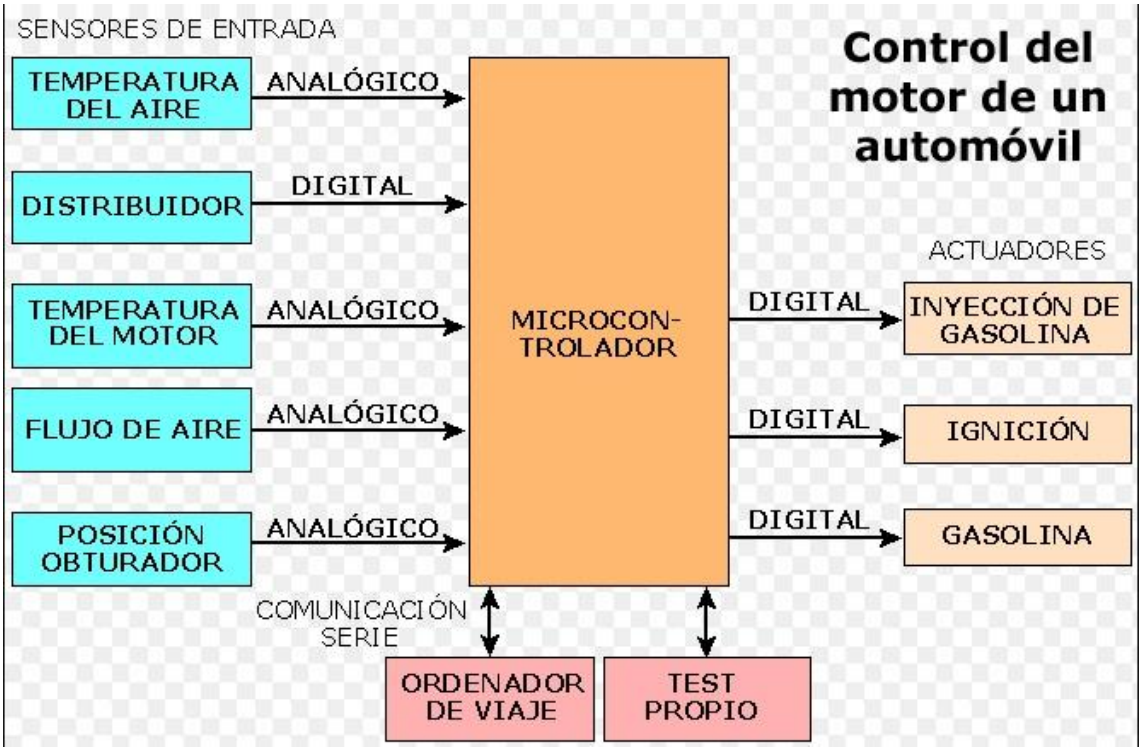
1C) Quais os principais periféricos de um Microcontrolador, descreva apliocação e utilização de periféricos.

Principais periféricos de um uC:



Aula 01 - Ricardo

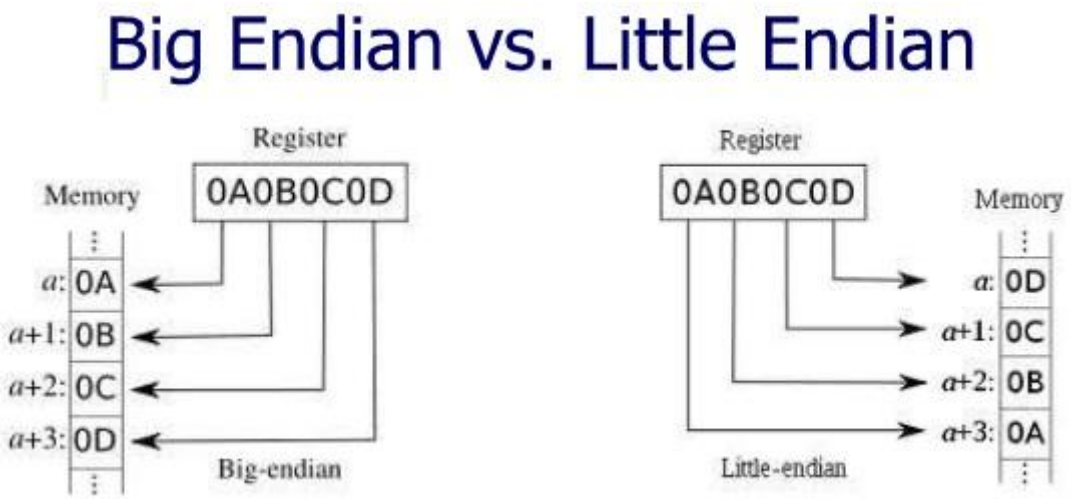
Exemplo Aplicação



1D) O que é Bigendian/Little Endian (endianness) ?

RESPOSTA:

Arquiteturas de armazenamento de memória de manipulação, relacionados á ordem em que os bytes são armazenados na memória.



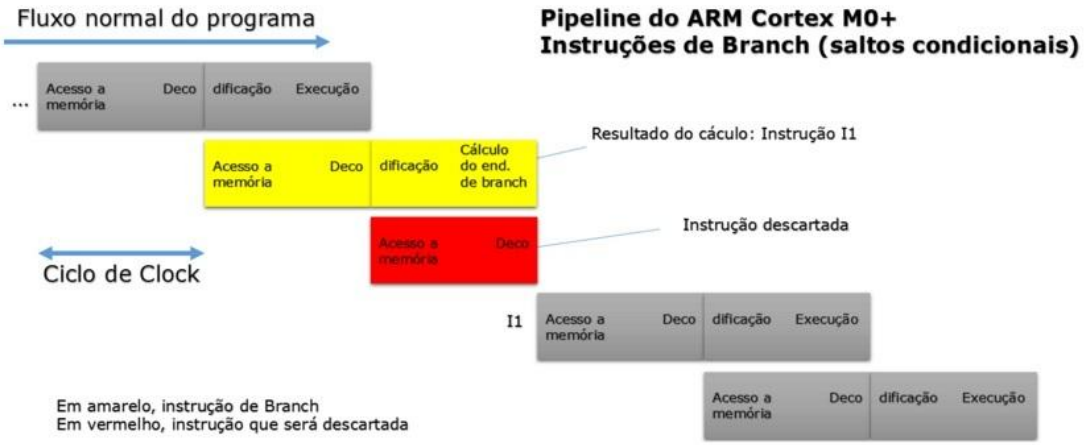
ARM

2B) O que é um Pipeline de um uC ?

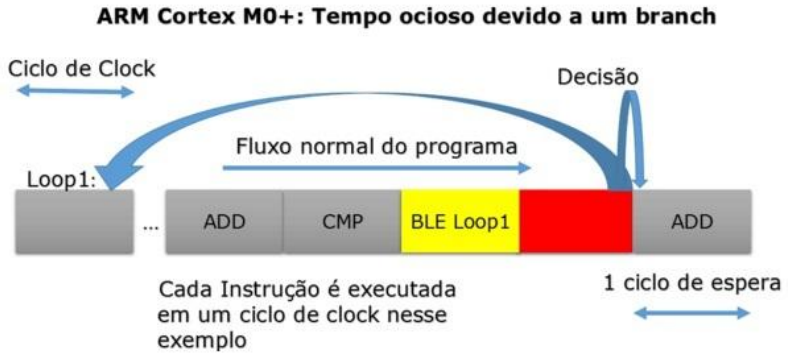
RESPOSTA:
Pipeline é uma técnica de *hardware* que permite que a CPU realize a busca de uma ou mais instruções além da próxima a ser executada.
Este tipo de arquitetura permite que, enquanto uma instrução é executada, uma outra seja "buscada" na memória, ou seja, um PIPELINE (sobreposição), o que torna o processamento mais rápido.

Cortex M0+: Dois estágio de *pipeline*

Essa arquitetura apresenta uma novidade: opera com dois estágios de pipeline apenas. Essa característica permite que o processador acesse menos a memória flash, e, assim, permite **menor consumo**, se comparado com uma mesma atividade do núcleo Cortex M0, que tem três estágios de *pipeline*, tal como os núcleos Cortex M3 e o ARM7.



As sombras de branch também são reduzidas e isso propicia um tempo de chamada à função bem pequeno e interrupções com latência menor.



Topicos Extras

3A) Qual a forma de medir desempenho de um microcontrolador

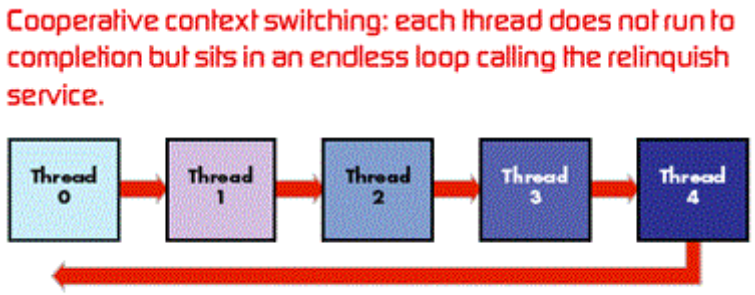
Utiliza-se software de stress como CoreMark

- Processamento básico
- Troca de contexto cooperativo
- Troca de contexto preemptivo
- Processamento de interrupção
- Processamento de interrupção com preempção
- Passagem de mensagem
- Processamento de semáforo
- Alocação e desalocação de memória

recomendação: operar a partir da memória Flash, que representa a condição mais significativa e desafiadora para medir o desempenho de uma arquitetura

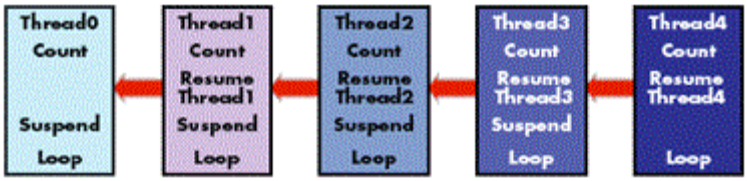
Cada um dos testes consiste de:

Troca de contexto cooperativo: Mede o tempo que o escalonador de um RTOS leva para trocar o contexto de execução de uma tarefa para outra de mesma prioridade. Para tanto 5 tarefas são criadas com a mesma prioridade. Cada uma das 5 tarefas executa um laço infinito chamando o serviço de desistência do processador.



Troca de contexto preemptivo: Mede o tempo que um RTOS leva para trocar o contexto de execução de uma tarefa para uma tarefa de maior prioridade através de preempção. Para tanto 5 tarefas com prioridades diferentes são criadas. A tarefa executa até ser preemptada por uma tarefa de maior prioridade. Todas as tarefas iniciam suspensas (menos a de menor prioridade). A tarefa de menor prioridade acorda a tarefa de mais alta prioridade seguinte, e assim por diante, até que a tarefa de maior prioridade seja executada. Cada tarefa incrementa seu contador de execução e se suspende. O processo é reiniciado quando o processador retorna para a tarefa de menor prioridade.

Preemptive context switching: each thread resumes a higher-priority thread, causing that higher-priority thread to be run, preempting the lower-priority thread and causing a context switch.

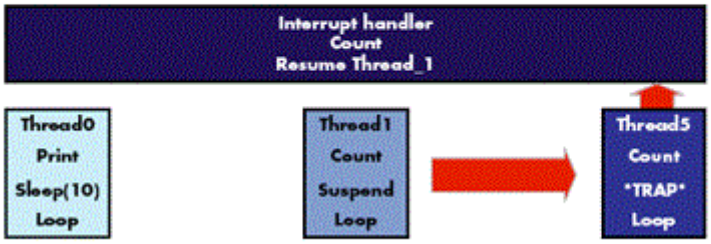


Processamento de interrupção: é o tempo necessário para saltar para uma interrupção, executar a rotina de tratamento desta interrupção, executar o escalonador para determinar qual tarefa deverá ser executada, restaurar o contexto desta tarefa (caso necessário) e saltar para a execução desta tarefa. Neste teste deve ser utilizado uma interrupção de *software* para realizar a preempção.
Teste sem preempção: Neste teste não há preempção, ou seja, a tarefa interrompida volta a ser executada. O tempo total medido é o tempo em que uma tarefa não está sendo executada. Muitos RTOSes anunciam / fazem propaganda de baixa latência de interrupção, mas deixam a desejar na latência das tarefas. Ambas as latências são críticas, sendo o tempo total o que realmente importa. O teste realizado considera a latência de interrupção e a sobrecarga para ativação das tarefas.

Interrupt processing: Thread 1 suspends itself allowing Thread5 (a lower-priority thread) to run. An interrupt is created as Thread5 executes a trap instruction [machine dependent]. The interrupt handler runs a different thread rather than a resume the interrupted thread. That involves a context save of the interrupted thread and a restore of the context for the new thread.

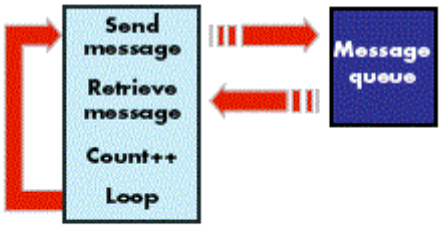
Teste com preempção: mede o tempo que um RTOS leva em situações onde uma tarefa diferente é selecionada pelo escalonador para executar na saída de uma interrupção. Este caso ocorre quando o tratamento de uma interrupção adiciona a lista de prontos uma tarefa de maior prioridade que a tarefa que estava sendo executada. O tempo para executar o salvamento de contexto da tarefa interrompida e restaurar o contexto para uma nova tarefa é incluído na pontuação deste teste. Para realizar este teste a tarefa 1 se suspende, permitindo que a tarefa 5 execute. A tarefa 5 força uma interrupção por *software*. O tratamento da interrupção acorda a tarefa 1. O escalonador troca o contexto da tarefa 5 para a tarefa 1 e a tarefa 1 é executada.

Interrupt processing: Thread 1 suspends itself allowing Thread5 (a lower-priority thread) to run. An interrupt is created as Thread5 executes a trap instruction (machine dependent). The interrupt handler resumes Thread 1, and the scheduler switches context to the higher-priority Thread 1.



Teste de passagem de mensagem: mede o tempo que um RTOS leva para passar uma mensagem de 16-bytes por valor, ou seja, copiada da origem para uma fila e recebida por uma tarefa. Mensagens maiores do que 16 bytes geralmente são passadas por referência (o ponteiro é enviado ao invés dos dados). Para realizar este teste uma tarefa envia uma mensagem de 16 bytes para uma fila e retira desta fila os mesmos 16 bytes. A tarefa incrementa um contador depois de enviar e receber a seqüencia completa de dados.

Message passing: a message is sent to a queue, then retrieved from the queue in an endless loop.

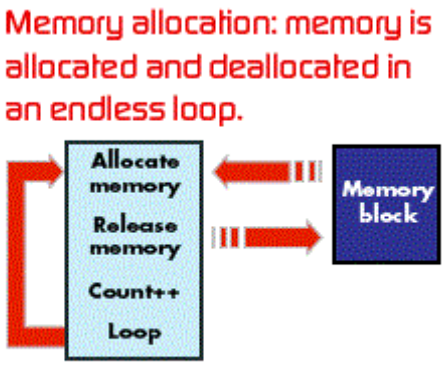


Teste de processamento de semáforo: Mede o tempo que um RTOS leva para "obter" e "liberar" um semáforo. Para realizar este teste uma tarefa obtém um semáforo e imediatamente o libera. Depois deste procedimento a tarefa incrementa o contador de execução.

Semaphore processing: a thread gets a semaphore and then releases it in an endless loop.

Teste de alocação de memória: Mede o tempo que leva para um RTOS alocar um bloco de memória de tamanho fixo (128 bytes) para uma tarefa. Se um RTOS não possuir gerenciamento de blocos de memória a função **malloc** do Ansi C deverá ser utilizada (provavelmente obtendo baixo desempenho). Para realizar este teste uma tarefa aloca um bloco de

memória de 128 bytes e logo em seguida desaloca este mesmo bloco. Depois de liberar o bloco de memória a tarefa incrementa o contador de execução.



3C) Classifique os Tipos de memória de um microcontrolador

- Normal
- Device(Dispositivo)
- Strongly-ordered (Fortemente-ordenada)

Na arquitetura ARMv7-M, tal como implementada nos processadores Cortex-M3 e Cortex-M4, e na arquitetura ARMv6-M, tal como implementada nos processadores Cortex-M0 e Cortex-M0 +, existem três tipos de memória mutuamente exclusivos especificados. Esses são: Normal Dispositivo Fortemente-ordenada. Normalmente, a memória utilizada para o programa e para armazenamento de dados é a memória normal. periféricos do sistema (I / O locais), geralmente em conformidade com diferentes regras de acesso à memória Normal. Exemplos de acessos de E / S são: registos controlador de interrupção em que um acesso pode ser usado como uma interrupção reconhecem, alterando o estado do controlador registros de configuração do controlador de memória que são usados para configurar o timing e correção de áreas de memória normal memória periféricos mapeados, onde o acesso a um local de memória pode causar sistema efeitos colaterais. Em ARMv7 incluindo ARMv7-M, e em regiões ARMv6-M do mapa de memória, os acessos aos periféricos do sistema são definidas como Dispositivo ou memória Fortemente-ordenada acessos, e são mais restritivos do que os acessos à memória Normal. Isso é: ler e escrever pode causar sistema efeitos colaterais acessos não deve ser repetido, por exemplo, no retorno de uma exceção o número, ordem e tamanhos de acessos deve ser mantida. Dentro de cada tipo de memória, as regiões podem ainda ser divididos através da atribuição de diferentes atributos de memória, tais como, a permissão de acesso e cacheability.

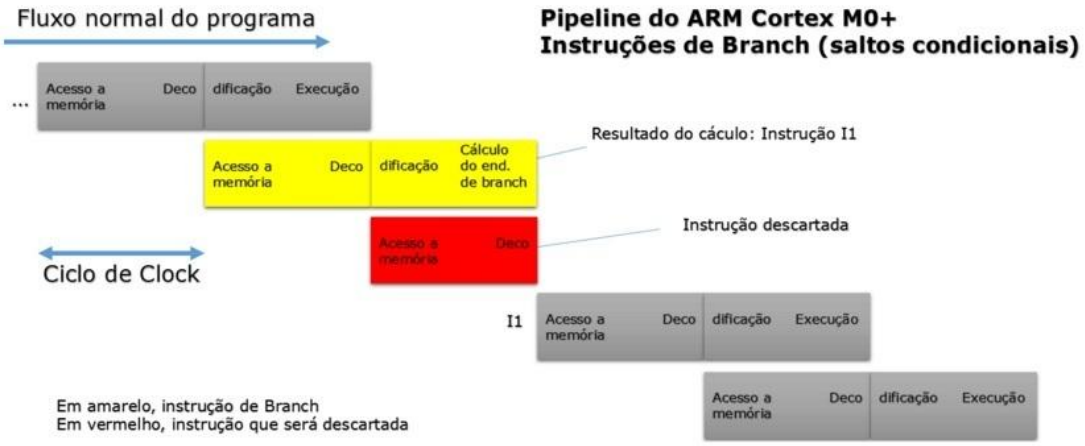
3D) Qual a diferença entre os tipos de variável Int, Chart, Float, Real

resp:

Int - número inteiro
Chart - string de caracteres
Float - número fracionário
Real - número real (maior limite de grandeza que o Float)

Cortex M0+: Dois estágio de *pipeline*

Essa arquitetura apresenta uma novidade: opera com dois estágios de pipeline apenas. Essa característica permite que o processador acesse menos a memória flash, e, assim, permite **menor consumo**, se comparado com uma mesma atividade do núcleo Cortex M0, que tem três estágios de *pipeline*, tal como os núcleos Cortex M3 e o ARM7.



As sombras de branch também são reduzidas e isso propicia um tempo de chamada à função bem pequeno e interrupções com latência menor.