

Sistema embebido - Sistema de Computação com uma função específica, dedicada, num sistema maior, mais complexo.

Microcontroladores - basicamente um computador dedicado utilizado em sistemas embebidos: Processador + Memória + Periféricos
Fatores de desempenho - Computadores – velocidade de processamento

Microcontroladores - Rapidez de resposta a eventos externos (Sistemas de Tempo Real RTS) – Consumo de potência – Minimização das necessidades de memória – Velocidade de processamento

AC2 I O 1.pdf

Os periféricos têm diferentes velocidades de operação e as suas velocidades de transferência são muito menores do que o processador e memória. Para isso usa-se **I/O modules** entre o processador e os periféricos: Que têm a função de controlar e temporizar (coordenação do fluxo de dados entre o sistema e os periféricos)

Comunicar com o processador e com o periférico. Buffer de dados (memória local) e detecção de erros.

Read e Write em relação ao processador. Existe um **Data register, status register e control register**.

Memory mapped I/O e I/O isolado

I/O programado, Interrupt-driven, DMA

AC2 I O 2.pdf

No MIPS não há prioridades de hardware.

No PIC32 as prioridades são controladas pelo Interrupt Controller externo ao processador.

PIC32 até 5 fontes de interrupção externas

96 fontes de interrupção

7 níveis de prioridade

7*4 (subníveis) de prioridade

Ligações ponto-a-ponto full crossbar (ligação entre cada 2 componentes através de linhas dedicadas)

Shared Bus as linhas podem ser partilhadas entre cada componente ou exclusivas para cada componente

Bus dedicado linhas distintas para endereços e dados

Bus multiplexado endereços e dados usam as mesmas linhas (os de control são sempre distintos)

Se o sistema tiver um único BUS e tiver um grande número de dispositivos ligados o desempenho degrada-se. O tempo de propagação dos sinais aumenta e pode ser o estrangulamento do sistema quando as transferências entre dispositivos aproximam-se do máximo suportado por ele.

Assíncrono os módulos respondem no seu próprio ritmo e fica mais fácil gerir os periféricos com diferentes tempos de resposta. É feito com tempo de resposta Handshake ficando assim mais lento porque implica 2 tempos de propagação de sinais no BUS.

Síncrono os módulos de interface (I/O) periféricos têm de ser capazes de responder no tempo de ciclo válido ficando assim condicionado a taxas de transferência no bus.

Árbitro centralizado tem um árbitro

Descentralizado todos os masters envolvidos não havendo árbitro individualizado

Non pre emptive uma vez obtido o controle do bus o módulo mantém até terminar

Pre emptive uma transferência de prioridade mais baixa é interrompida por um pedido de acesso de um módulo com prioridade elevada

Round robin o acesso ao bus é atribuído rotativamente

Starvation transf críticas podem ter de esperar um tempo longo

Split transfers quando os dispositivos que estão ligados ao master necessitam de vários ciclos de relógio para colocar no bus um dado então o bus que estava atribuído ao master permanece bloqueado para os outros masters não estando este a ser usado o árbitro sabe quando isto acontece e então atribui o bus a outro master quando o slave estiver pronto então indica ao árbitro e o slave completa a transferência

DMA por bloco em bloco, o DMA assume o controlo do bus que pertencia ao CPU e transfere todos os dados

DMA por burst o DMA transfere até atingir um número programado de palavras ou até o periférico não ter mais informação pronta a ser transferida

PCI bus é **síncrono** tem relógio para controlar todas as transferências e os módulos de interface vão ter de ser capazes de responder no tempo de ciclo válido. E é **multiplexado** porque usa linhas comuns para os endereços e dados. **Arbitragem centralizada** os módulos I/O obedecem a um árbitro.

PCI express tem bus partilhado mas as ligações ponto a ponto estão a tornar-se comuns. o **bus partilhado** tem alguns problemas: limitações física ao aumento da frequência de relógio, sincronização e arbitragem tornam-se difíceis de efectuar em tempo útil. **vantagens de ligação ponto a ponto** menor latência, maiores velocidades de transferência.

Uma **UART** é uma Universal Asynchronous Receiver Transmitter, transmite de forma sequencial. Ao receber junta todos os bits em bytes usando um conversor série-paralelo. Os sinais da UART são convertidos por dispositivos de interface para as diferentes tecnologias.

Simplex apenas numa só direção **Half-duplex** alternadamente **FullDuplex** em ambas as direções

RS232 permite a comunicação bidirecional full duplex. Usa a mesma estrutura da UART, start bit de 1 para 0, podem ser usados 1 ou 2 bits de stop bits. A definição do parity bits também é essencial.

Uma **UART** é uma Universal Asynchronous Receiver Transmitter, transmite de forma sequencial. Ao receber junta todos os bits em bytes usando um conversor série-paralelo.

Os sinais da UART são convertidos por dispositivos de interface para as diferentes tecnologias.

USB

- No RS232 e UART o utilizador tem de configurar todo o processo. O mesmo já não acontece no USB, que permite a ligação de periféricos a computadores de um modo standard, com um protocolo standard. Os dispositivos interagem com o sistema de operação sem intervenção do utilizador.

- Permite a ligar e retirar dispositivos a qualquer altura sem necessidade de reiniciar o sistema.
- Pode fornecer energia aos periféricos através do mesmo cabo utilizado para a comunicação de dados.
- Permite ligar periféricos com necessidades de taxas de transferências de dados muito mais rápidas.
- Permite ligar periféricos com grandes necessidades de taxas de transferência.

Um bus **I2C** é um **BUS série, multi-master e half-duplex** (biderecional, serial data line, serial clock line).

SPI - Full-duplex biderecional, comunicação feita em **4 linhas**. Tem apenas um **único BUS master**. Arquitectura master-slave com ligação **ponto-a-ponto**.

Funciona como data-exchange. Para cada bit que é enviado para o receptor também é recebido um bit no master. Isto é, ao fim de N ciclos de relógio o transmissor enviou uma palavra de N bits e recebeu outra com a mesma dimensão.

Controller Area Network

Vantagens do CAN

- Standard bem definido
- Existência de muitos produtos e ferramentas CAN no mercado
- Protocolo implementado em hardware
- O tratamento de erros e transmissão são bem definidos e têm velocidades elevadas
 - CRC para detetar erros
 - mecanismo para evitar que um nó avariado bloqueie o sistema
- Muito usado na indústria e em veículos de transporte
- A melhor relação preço/ performance
- Meio de transmissão simples
 - Twisted pair
 - ou um fio único

Características

- Bus multi-master, assíncrono
- Inexistência de endereçamento de nós
- Broadcasting das mensagens (todos recebem as mensagens)
- É uma rede fechada, não é preciso logins ou interface com o user

CANopen

Características

- CANopen é um subconjunto de CAN Application Layer
- Tem auto-configuração de rede
- Acesso fácil a todos os parâmetros dos dispositivos
- Sincronização dos dispositivos
- Transferência de dados cíclica e desencadeada por eventos
- Estabelecimento de leitura inputs, outputs ou parâmetros síncrona

Protocol

- Cada nó é recetor e transmissor
- Cada mensagem é transmitida a todos os dispositivos ligados ao bus (broadcast)
- Todos os nós lêem a mensagem, e decidem se é relevante para eles
- Todos os nós verificam a inexistência de erros na recepção
- Os nós fazem o acknowledge da recepção

