

Università di Ferrara
Facoltà di Ingegneria



Docente: Ing. Massimiliano Ruggeri

Mail: m.ruggeri@imamoter.cnr.it

mruggeri@ing.unife.it

Tel. 0532/735631

Corso di Sistemi di Elaborazione A.A. 2008/2009

Durata: circa 60 ore.

Orario: lunedì 3 ore (8:30->11:30), giovedì 2 ore (h8:30 -> 10:30), giovedì 2 ore (h 8:30 -> 10:30);

Inizio: 06 aprile 2009; termine: XX giugno 2009

Prova d'esame: prova scritta / ricerche?

Frequenza degli esami:

Scritto: due - tre per sessione

Normalmente in coincidenza degli scritti di calcolatori

Obiettivi del corso

Introdurre alla **progettazione di schede hardware** basate su microprocessori e microcontrollori

Presentare i principali metodi di **interfacciamento** tra processore e **periferiche** e le metodologie più comuni di **gestione delle periferiche** stesse, inclusa l'analisi di semplici driver in C standard.

Presentare i principali **protocolli di rete seriale** utilizzati sia in ambiente embedded che general purpose e i **circuiti hardware** che le realizzano

Presentare i principali problemi di interfacciamento e scambio dati in **sistemi multiprocessore**.

Organizzazione del corso

Propedeuticità consigliate:

Calcolatori Elettronici, Programmazione in linguaggio C

Laboratorio per il modulo di Sistemi di Elaborazione:

Esercitazioni facoltative su schede a microprocessore/microcontrollore e schede CAN, sistemi analizzatori di rete e di protocollo

Testi consigliati:

Dispense del corso redatte a cura del docente, integrate da data sheet dei dispositivi.

Orario di ricevimento:

t.b.d.

Programma del corso

1. Architettura a livello di sistema (con riferimento ai processori Microchip PIC18F8x20 e ARM7/9)

Architettura a livello di sistema. Interfaccia standard nei microprocessori.

I segnali esterni dei microprocessori PIC18F8x20.

Architettura dei bus multiplexati: bus di memorie, di I/O. Circuiti logici di interfaccia di bus. Il bus esterno del PIC18: Interfaccia in Microprocessor Mode, Extended Microprocessor Mode, Microprocessor Mode with Boot Block..

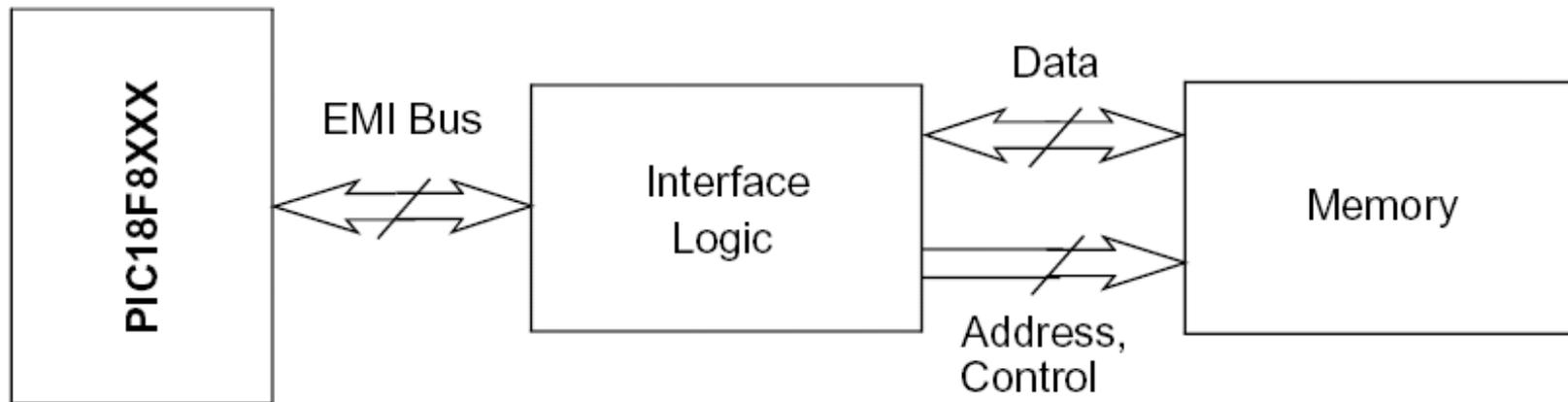
Confronto con i micro ARM.

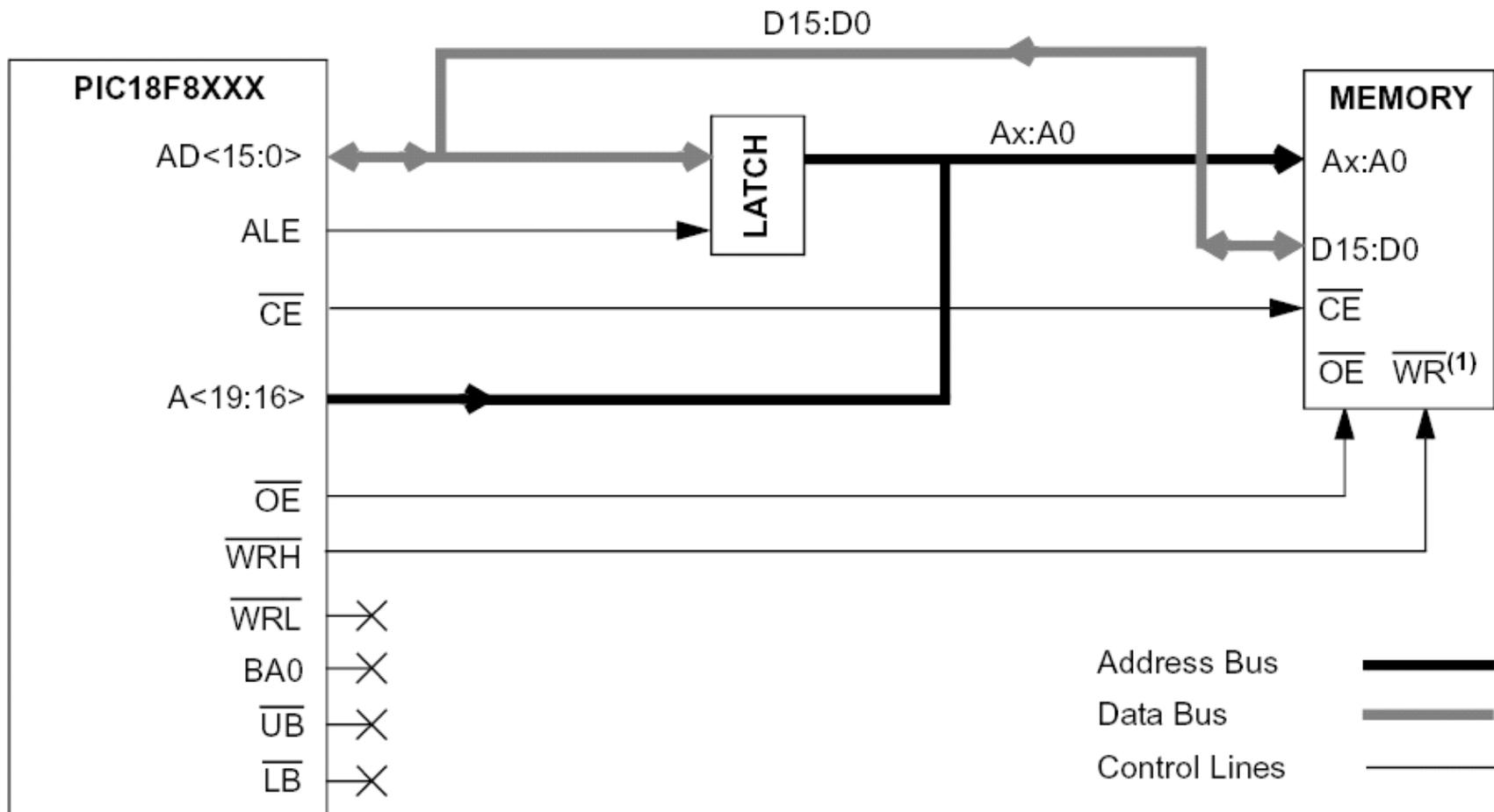
Gestione del reset, del clock. Cicli di wait.

Gestione delle periferiche integrate nei microprocessori PIC18F8x20, inizializzazione del micro, set-up delle periferiche e programmazione in C embedded (relative limitazioni).

2. Interfaccia standard con memorie

Cicli di lettura e scrittura da e per memoria. Cicli di fetch. Temporizzazioni dei segnali del PIC18F8x20 nei cicli di fetch, di table read e table write (lettura e scrittura di operandi). Progetto di sottosistemi di memoria (lettura e scrittura di word in uno o due cicli di bus, e di byte in un ciclo di bus).





3. Porta parallela e interfaccia con le periferiche su porta parallela

Architettura a livello di sistema. Periferiche e CPU. La porta parallela 8255. Trasferimenti dei dati attraverso porta parallela. Protocollo Basic I/O e Strobed I/O con handshake asincrono.

Porte di I/O e Bus esterno.

Compact Flash (International standard) Interfacciamento e driver in C.

Esempi di uso ed esercizi.

4. Gestione delle interruzioni

Modalità di comunicazione tra CPU ed I/O: polling ed interrupt. Gestione dell'interrupt dal punto di vista software. Protocollo di risposta all'interrupt nel PIC (sorgente interna), Espansione con Controller esterno, ricostruzione del protocollo di risposta all'interrupt delle macchine Intel.

L'interrupt controller 8259. Gestione della priorità ed interrupt in cascata. Esempi di progetto.

La gestione degli Interrupt nei microprocessori PIC, driver in C.

5. Gestione dell' accesso diretto alla memoria

Sistemi multi-master. Interfaccia tra il processore e un dispositivo master. Protocollo HOLD/HOLDA e RQ/GT. Trasferimento dati da e per memoria. Il Direct Memory Access: macchina a stati finiti. Trasferimenti Fly by e Flow Through. Il DMA Controller 8237. Progettazione del bus di sistema in ambiente multimaster. Memory Mapped I/O. Esempi di progettazione di interfaccia tra DMAC, CPU e memorie. Esempi di trasferimento dati dalla porta parallela in DMA.

6. Contatori programmabili

I Timer interni del PIC18F8x20.

Il Timer esterno 8254. Modalità di conteggio. Uso del timer come contatore di eventi asincroni e come divisore di frequenza. Interfaccia con la CPU polling e ad interrupt. Driver del contatore. Esempi di progetto.

Uso dei Timer e funzionalità di Input Capture, Output Compare e generazione di forme d'onda PWM.

7. Interfacce seriali

Interfaccia seriale asincrona. Standard RS-232. La periferica UART dei PIC18F8x20. Trasferimento asincrono con interrupt o con DMA. Esempi di progetto.

Rete seriale LIN in ambiente automotive.

Interfacce seriali sincrone, comunicazione all'interno delle schede di controllo su bus seriale: il bus I2C e il bus SPI.

La periferica MSSP (Master Synchronous Serial Port) del PIC18F8x20.

Interfaccia con dispositivi seriali sincroni (memorie, RTC, Dispositivi di potenza, Shift register)

8. Il bus Universal Serial Bus (USB)

Il bus Universal Serial Bus (USB): concetti, struttura, tipi di trasferimento, tipi di pacchetti. Esempi di comunicazione. Banda passante utile per i dati.

Il controllore USB Philips PDIUSB12: blocchi funzionali e pin-out; modalità di funzionamento; temporizzazioni di lettura e scrittura. Il controllore USB D12 in modalità in DMA. Compatibilità tra il D12 e il DMA controller 8237A-5. Programmazione.

Il controllore USB Philips PDIUSB11: USB da interfaccia SPI: driver in C.

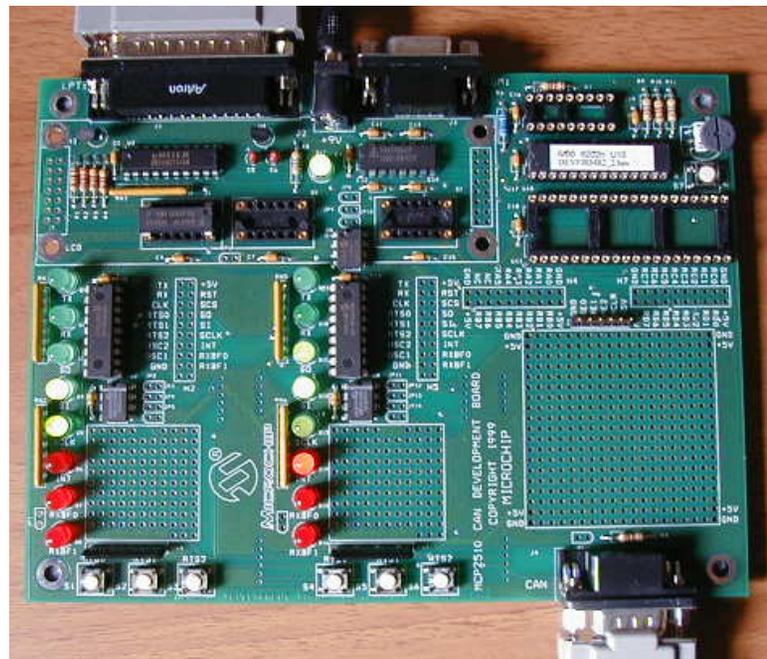
Cenni a bus seriali più evoluti: IEEE 1394.

9. Il bus CAN

Il bus CAN (Controller Area Network) Bosch Automotive Group: concetti, struttura, tipi di trasferimento, tipi di pacchetti. Esempi di comunicazione. Banda passante utile per i dati. Esempi di reti CAN.

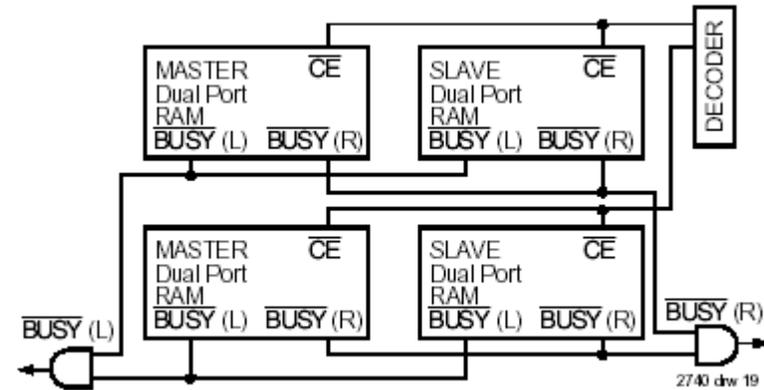
Il controllore di rete CAN microchip MCP2510: blocchi funzionali, struttura e modalità di programmazione. Esempi di funzionamento.

L'interfaccia SPI per il controllore MCP2510: driver in C.



10. Dual Port RAM e sistemi multiprocessore

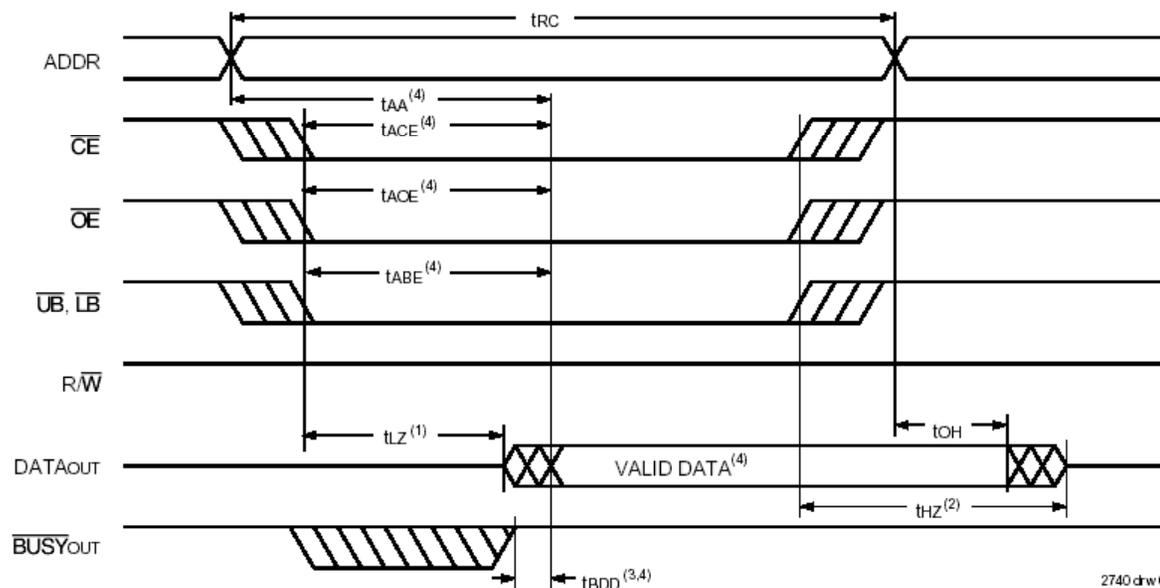
Il problema dell'interfacciamento, della validità dei dati e dell'accesso alla memoria, gestione di memorie condivise



IDT7024S/L
High-Speed 4K x 16 Dual-Port Static RAM

Military, Industrial and Commercial Temperature Ranges

Waveform of Read Cycles⁽⁵⁾



NOTES:

1. Timing depends on which signal is asserted last, \overline{CE} , \overline{OE} , \overline{LB} , or \overline{UB} .
2. Timing depends on which signal is de-asserted first, \overline{CE} , \overline{OE} , \overline{LB} , or \overline{UB} .
3. t_{BDD} delay is required only in cases where opposite port is completing a write operation to the same address location. For simultaneous read operations \overline{BUSY} has no relation to valid output data.
4. Start of valid data depends on which timing becomes effective last t_{ABE} , t_{AOE} , t_{ACE} , t_{AA} or t_{BDD} .
5. SEM = Vih.

11. L'Interfaccia JTAG

Lo standard JTAG per il debug di microcontrollori/microprocessori, DSP e FPGA.

TESI

- Progettazione schede a microprocessore/microcontrollore per ambiente embedded.
- Progettazione schede per interfacciamento sistemi in rete USB e CAN.
- Reti CAN / CAN triggered.
- Diagnostica su rete GSM/GPRS.
- Data logger con Compact Flash/USB.
- Dashboard con LCD grafici.
- Sistemi multiprocessore aderenti alle normative di safety per il mondo avionico (IEC61508)
- Sistemi embedded a PS con sistemi operativi real time