

Laurea Magistrale in Ingegneria Informatica
Laurea Magistrale in Ingegneria Elettronica e Telecomunicazioni per
lo Sviluppo Sostenibile

Sistemi e Tecnologie per l'Automazione LM

HW per elaborazione digitale in automazione: Microcontrollori e DSP

Ing. Gianluca Palli
DEI - Università di Bologna
Tel. 051-2093186
E-mail: gianluca.palli@unibo.it
<http://www-lar.deis.unibo.it/people/gpalli/>

Revisionato il 07/10/2015

Unità di controllo digitali

Introduzione

■ Varie tipologie di soluzioni

- 1: Elaboratori **general purpose** (o quasi)
 - Basate su **PC standard o industriali** + opportune I/O board
- 2: **Custom**
 - Due livelli possibili:
 - ◆ Costruttore di **unità di controllo general purpose** (es: PLC, vedi in seguito)
 - ◆ Costruttore di **sistemi di controllo/automazione**
- (Eventualmente distribuite)

■ Scopo di tale sezione: Introdurre alcuni processori particolari usati nelle unità di controllo ed evidenziarne le caratteristiche

- Microcontrollori (μC) e Digital Signal Processors (DSP)
- Usati in architetture custom

Unità di controllo digitali "custom"

Caratteristiche generali

■ Uso di processori speciali

- Microprocessori 8/16/32 bit
- Microcontrollori
- Processori di segnale (DSP)
- Field-Programmable Gate Array (FPGA)
- Application Specific Integrated Circuit (ASIC)
 - non riprogrammabili

■ programmazione:

- gestione dell'HW di scheda e delle funzionalità di base come le temporizzazioni (**Real-Time** - vedi avanti) (**S.O. di scheda**: molto semplice \Rightarrow kernel)
- Procedure di Utente (Algoritmi di controllo)



MicroControllore (μ C)

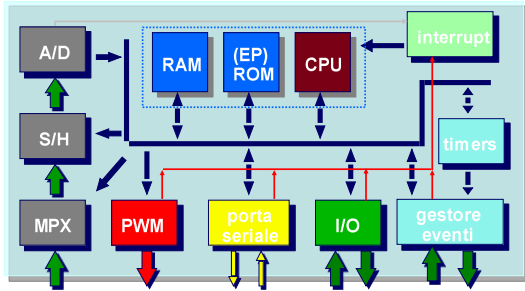
MicroControllore

- Speciale microprocessore per il controllo, quindi orientato a:
 - **grande capacità di gestione I/O e Tempo**
 - **ottimizzazione per ridurre i costi e gli ingombri**
 - risorse hardware a bordo
- Spesso realizzato su specifica per ottimizzare un prodotto o una classe di prodotti
 - **stampanti, automobile, telefonia**

MicroControllore

- **Integrazione nel chip di periferiche tipiche dei sistemi di controllo**
 - **memoria** (EP)ROM, RAM
 - **convertitore D/A** modulatore PWM
 - **convertitore A/D** n canali + multiplexer
 - **interfaccia di comunicazione seriale**
 - **Sincrona (SPI) / asincrona (SCI)**
 - **Eventuali interfacce per bus di campo**
 - ◆ **In forte diffusione l'interfaccia CAN**
 - ◆ **In forte diffusione anche l'Ethernet controller**
 - **gestione di tempo ed eventi**
 - **timers, watch-dog, gestore interrupt**
 - **porte di I/O digitale**
 - **configurabili singolarmente**

MicroControllore



CPU (= ALU+Registri) è il tipico μ P

MicroControllore

Un semplice esempio Arduino



MicroControllore (Arduino)

■ Caratteristiche generali

- ▮ Arduino è un microcontrollore totalmente open source (hardware e software)
 - pagina web: www.arduino.cc
 - basso costo
 - applicazioni principalmente nel mondo della robotica e sistemi em bedded
 - ◆ linguaggio di programmazione di "alto livello" (simile al C)
 - ◆ supporto a operazioni di I/O per interfacciare il sistema con sensori e attuatori
 - basato su microcontrollori a 8 bit
 - ◆ varie schede basate su microcontrollori Atmel (es. ATmega328, ATmega2560, ...)
 - ◆ versioni future (Arduino 2) sono basate su processori ARM a 32 bit
- ▮ ARDUINO Shields
 - IDEA: mantenere un formato standard di scheda per permettere espandibilità attraverso HW aggiuntivo

MicroControllore (Arduino)

Hardware (Arduino MEGA 2560)

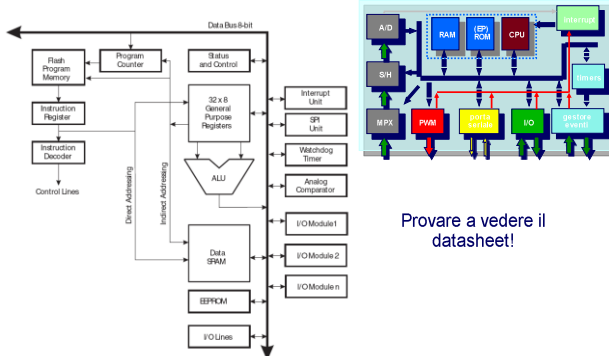
■ Basato sul microcontrollore Atmel ATmega2560 a 8 bit

Device	Flash	EEPROM	RAM	General Purpose I/O pins	16 bits resolution PWM channels	Serial USARTs	ADC Channels
ATmega640	64KB	4KB	8KB	86	12	4	16
ATmega1280	128KB	4KB	8KB	86	12	4	16
ATmega1281	128KB	4KB	8KB	54	6	2	8
ATmega2560	256KB	4KB	8KB	86	12	4	16
ATmega2561	256KB	4KB	8KB	54	6	2	8

- 4 porte seriali
- 16 ingressi analogici (10 bit di risoluzione)
- 12 uscite di tipo PWM
- 86 I/O pins configurabili
 - comunicazioni digitali (I2C, SPI, etc.), interruttori per funzioni logiche, etc.

MicroControllore (Arduino)

■ Architettura Atmel ATmega2560

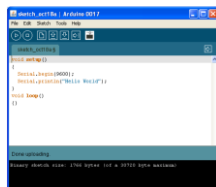


MicroControllore (Arduino)

Ambiente Arduino

■ Ambiente di programmazione

- si può scaricare gratuitamente da
 - <http://arduino.cc/en/Main/Software>



- guide al linguaggio ed esempi sono disponibili nella reference
 - <http://arduino.cc/en/Reference/HomePage>
- esistono simulatori che permettono di verificare ed eseguire virtualmente il codice senza avere ancora a disposizione l'hardware
 - <http://virtronics.com.au/Simulator-for-Arduino.html> (free fino a 100 righe di codice)

Digital Signal Processor (DSP)

Digital Signal Processor (DSP)

Processore progettato per minimizzare i costi della elaborazione digitale dei segnali:

⇒ REALIZZAZIONE CONTROLLI TEMPO DISCRETO

- Nato nei primi anni 80 in Elettronica/Telecomunicazioni
 - Usato anche come co-processore matematico
 - Signal processing
 - ◆ Audio-Video digitale
 - ◆ Filtraggio/Analisi segnali in comunicazioni
- Nell'Automazione: Controllo diretto di variabili temporali
 - Ad elevata complessità
 - Anche non lineare
 - ◆ Azionamenti elettrici
 - ◆ Controllo coordinato del moto
 - ◆ Robotica
 - ◆ Controllo di assetto
 - ◆

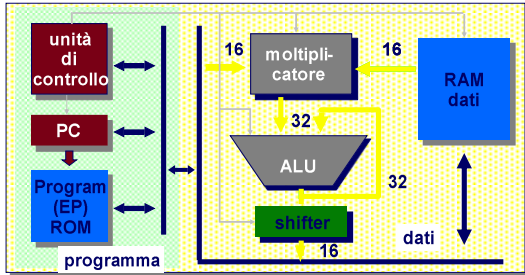
Digital Signal Processor (DSP)

NOTA BENE

- Prestazioni in elaborazione (MIPS/MFLOPS) dei DSP sono molto elevate
 - Fino a circa 5000 MIPS e 1000MFLOPS di picco
- Ma ottenibili e superabili con processori da PC
- Però con costi/ingombri e consumi molto maggiori
 - DSP: frequenze di clock sotto 1GHz

Digital Signal Processor (DSP)

Architettura del Chip



Ing. Gianluca Palli - Sistemi e Tecnologie per l'Automazione LM Microcontrollori e DSP 19

Digital Signal Processor (DSP)

Architettura del Chip

- Moltiplicatore hardware integrato
 - ALU 16/32 bit
 - fixed e floating point
- Architettura Harward
 - bus dati e bus istruzioni separati
- Operazioni "al volo" (shift)
- Struttura a Pipe-line
 - Anche più di una
- Very Long Instruction Word
 - Fetch contemporaneo di molte istruzioni
- Tempo di istruzione: <10ns (anche <3ns)
 - Non eccelso (crf processori da PC attuali, però... vedi prima)
 - Cm q grazie a pipe line (anche multiple) ⇒ throughput > centinaia MIPS e MFLOPS

Ing. Gianluca Palli - Sistemi e Tecnologie per l'Automazione LM Microcontrollori e DSP 20

Digital Signal Processor (DSP)

Principali caratteristiche di esecuzione

- linguaggio assembler ottimizzato per l'esecuzione di prodotti scalari
 - moltiplicazione e somma (MAC) nello stesso ciclo macchina
 - shift di un dato in memoria durante la moltiplicazione
 - saturazione delle operazioni di somma (fixed-point)
 - gestione hw ottimizzata dei loop
 - gestione hw ottimizzata per operazioni su vettori e matrici
- tempo di esecuzione di un PID
 - 13 cicli macchina ▫ < 2µs tipico
- pre-fetch di più istruzioni e strutture a pipeline
 - problemi (tempi) di gestione branch e interrupt

Ing. Gianluca Palli - Sistemi e Tecnologie per l'Automazione LM Microcontrollori e DSP 21

Digital Signal Processor (DSP)

Sviluppi recenti

■ Versioni tipo microcontrollore

- **Periferiche a bordo**
- **Migliore gestione di branch e interrupt**
- **Detti Digital Signal Controller**
 - spesso indicati semplicemente come μC

In Automazione:

Dispositivo ideale per Ctrl diretto di variabili temporali ad alta complessità ed elevata dinamica

Qualche esempio

■ Alcune architetture (cenni):

- **ARM Cortex M family**
 - alcune applicazioni: 
- **TMS320**
 - DSP a 16 bit
 - ◆ alcune varianti integrano processori di tipo ARM (OMAP: ARM Cortex A core)



Field-Programmable Gate Array (FPGA)

Field-Programmable Gate Array (FPGA)

Field-Programmable Gate Array (FPGA)

⇒ Insieme di porte logiche (gate) le cui interconnessioni sono programmabili via software

⇒ Il programma modifica l'HW!!!

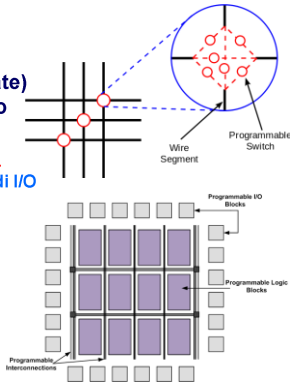
- Anche le funzioni dei pin di I/O sono programmabili

■ Vantaggi

- Performance
- Integrazione

■ Svantaggi

- Difficile programmazione
- Debug complesso
- Alee, propagazione del clock, ...



Field-Programmable Gate Array (FPGA)

⇒ Vantaggi

▫ Performance

- Le «istruzioni» sono implementate in HW -> tempo di esecuzione nell'ordine del ns
- Logica sequenziale con clock nell'ordine del GHz



▫ Integrazione

- Parallelismo reale
 - ◆ Le «istruzioni» sono eseguite realmente in parallelo
- Elevato numero di gate
- I/O riconfigurabile
 - ◆ Ottimizzazione del design della board



Field-Programmable Gate Array (FPGA)

⇒ Svantaggi

▫ Difficoltà di programmazione

- Il programma non definisce le operazioni da eseguire ma configura l'HW
 - ◆ Duale rispetto a programmazione convenzionale



▫ Difficoltà di debug

- Tempi di compilazione e riprogrammazione lunghi
- Limitata accessibilità dell'HW
- Uso di simulatori



▫ Problema delle Alee

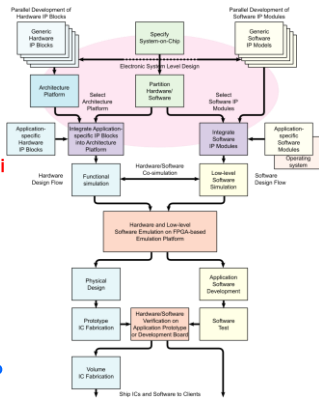
▫ Propagazione del Clock

Field-Programmable Gate Array (FPGA)

Flusso di progetto in ambiente FPGA - SoC

⇒ Separazione delle funzioni HW/SW

- Funzioni HW = Massimizzazione delle prestazioni vs. difficoltà di progetto e debug
 - Parallelismo reale
 - Scarsa flessibilità
- Funzioni SW = Programmazione convenzionale in C/C++
 - Facilmente riprogrammabili
 - Algoritmi di alto livello



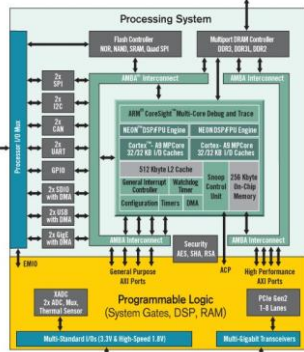
Ing. Gianluca Palli - Sistemi e Tecnologie per l'Automazione LM Microcontrollori e DSP 31

Field-Programmable Gate Array (FPGA)

Convergenza fra mondo Microcontrollori e FPGA

⇒ Zynq-7000

- Processore ARM Multicore
- Controllore DRAM, Flash
- Interfacce Seriali
 - UART (RS232, RS485)
 - SPI, I2C
- Bus di campo (CAN)
- Giga Ethernet
- Interfacce utente
 - USB, SDIO
- FPGA
 - Programmabile dall'utente
 - Interfacce ADC, PCIe



Ing. Gianluca Palli - Sistemi e Tecnologie per l'Automazione LM Microcontrollori e DSP 32

Laurea Magistrale in Ingegneria Informatica
Laurea Magistrale in Ingegneria Elettronica e Telecomunicazioni per lo Sviluppo Sostenibile

Sistemi e Tecnologie per l'Automazione LS

HW per elaborazione digitale in automazione:
Microcontrollori e DSP

FINE

Ing. Gianluca Palli
DEI - Università di Bologna
Tel. 051-2093186
E-mail: gianluca.palli@unibo.it
<http://www-lar.deis.unibo.it/people/gpalli/>