Progettazione di System-on-Chip (SoC)

Lezione 08 di Sistemi dedicati

Docente: Giuseppe Scollo

Università di Catania
Dipartimento di Matematica e Informatica
Corso di Laurea Magistrale in Informatica, AA 2018-19

Indice

argomenti della lezione

di che si tratta:

- il concetto di sistema su chip (SoC)
  - interrelazioni dei principali componenti
  - interfacce SoC per hardware custom
- principi di progettazione di SoC
  - elaborazione eterogenea e distribuita
  - comunicazione eterogenea e distribuita
  - memoria eterogenea e distribuita
  - controllo gerarchico
- esempio: un Soc multimediale
  - architettura del SoC
  - analisi del progetto

il concetto di sistema su chip

SoC: una *piattaforma* su singolo chip specializzata per un dominio applicativo

- il dominio applicativo influenza molto il tipo di periferia hardware, la dimensione delle memorie e la natura delle comunicazioni su chip
- un SoC è un sistema HW/SW specializzato e tuttavia *versatile*

esempi di dominio:

telefonia mobile, elaborazione video, comunicazione di rete ad alta velocità

esempi di applicazioni di elaborazione video:

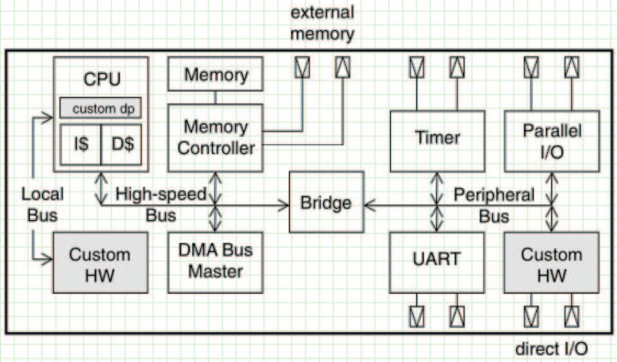
transcodifica di immagini, (de)compressione, trasformazioni di colori

vantaggi della specializzazione di dominio:

- *specializzazione* → maggiore efficienza di elaborazione
  → minor consumo di potenza o più alta velocità di elaborazione
- *versatilità* → riusabilità per molteplici applicazioni nel dominio
  → minor costo di progetto per applicazione e SoC più a buon mercato

interrelazioni dei principali componenti

quattro dimensioni ortogonali di analisi di organizzazione e interrelazioni dei componenti in un SoC:



controllo

elaborazione

comunicazione

memoria

Schaumont, Figure 8.1 - Generic template for a system-on-chip

in questa architettura generica:

- controllo di più alto livello da un microprocessore general-purpose (tipicamente RISC), funzioni di controllo specifiche da altri componenti
- elaborazione, comunicazione e memoria sono sia *eterogenee* sia *distribuite*

interfacce SoC per hardware custom

i blocchi grigi in fig. 8.1 mostrano tre modi per inserire hardware custom in un SoC

- come *periferica standard* su un bus di sistema
  l'approccio più generale: comunicazione via R/W memory-mapped I/O
  (+) meccanismo universale uniforme
  (-) bassa scalabilità, il bus di sistema diventa presto un collo di bottiglia
- come *coprocessore* su bus locale o interfaccia di coprocessore
  comunicazione attraverso un protocollo dedicato
  (+) maggiore larghezza di banda, minore latenza
  (-) dipendenza dal protocollo o dall'interfaccia di coprocessore fornita dal microprocessore
- come custom-hardware *datapath interno al microprocessore*
  estensione dell'insieme di istruzioni, comunicazione via banco dei registri
  (+) altissima larghezza di banda
  (-) forte dipendenza dal microprocessore e dai suoi colli di bottiglia

nella progettazione di SoC, non vi è un solo modo *migliore* di integrare hardware e software

alcuni fattori da bilanciare nel progetto di SoC:

- larghezza di banda richiesta per la comunicazione
- complessità di progetto dell'interfaccia hardware custom
- sviluppo del software
- tempo disponibile per il progetto
- budget di costo complessivo

quattro principi di progettazione per *qualsiasi* SoC:

- elaborazione eterogenea e distribuita
- comunicazione eterogenea e distribuita
- memoria eterogenea e distribuita
- controllo gerarchico

eterogeneità hardware: FSMD, macchine microprogrammate, microprocessori RISC

tutte possono realizzare il parallelismo al livello di micro-operazioni e di istruzioni, ma nessuna lo può davvero al livello di task, al quale sono macchine sequenziali

l'elaborazione parallela al livello di task è possibile in un SoC grazie alla loro *molteplicità*

eterogeneità funzionale: unità computazionalmente diverse

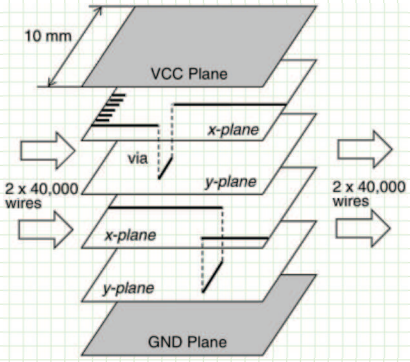
unità specializzate per applicazioni diverse nel dominio, e.g. analisi, compressione, elaborazione di immagini in un chip DSP

grazie al parallelismo a tutti i livelli un SoC può sfruttare pienamente la tecnologia hardware, sotto due aspetti:

- la specializzazione di ciascuna unità funzionale ne permette una realizzazione in silicio molto più efficiente di una in software su RISC general-purpose
- unità funzionalmente concorrenti possono operare effettivamente in parallelo

più *segmenti di bus* connessi da ponti possono evitare il collo di bottiglia del bus centrale

le alte variazioni dei requisiti di comunicazione su chip richiedono *eterogeneità* delle connessioni sul chip: bus condivisi, connessioni punto-punto, seriali, parallele



Schaumont, Figure 8.2 - Demonstration of the routing density in a six-layer metal 90 nm CMOS chip

un esempio dal fondatore di Tensilica, Chris Rowen sull'estremamente elevata larghezza di banda della comunicazione su chip:

ipotesi:

- processore a sei strati di metallo da 90 nm, con due coppie di strati per l'instradamento
- densità di linee: 4/µm, frequenza: 500 MHz

→ una larghezza di banda teorica di 40 Tbps!

la comunicazione off-chip è minore per ordini di grandezza

e.g. una connessione Hypertransport 3.1, uno standard per processori ad alta velocità, con 4 porte dà una larghezza di banda di ~200 Gbps

l'eterogeneità delle memorie su silicio in un SoC è riassunta in tabella 8.1

Type	Register Register file	DRAM	SRAM	NVROM (ROM, PROM, EPROM)	NVRAM (Flash, EEPROM)
Cell size (bit)	10 transistors	1 transistor	4 transistors	1 transistor	1 transistor
Retention	0	Tens of ms	0	∞	10 years
Addressing	Implicit	Multiplexed	Non-muxed	Non-muxed	Non-muxed
Access time	< 1 ns	< 20 ns	< 10 ns	20 ns	20 ns (read) 100 µs (write)
Power consumption	High	Low	High	Very low	Very low
Write durability	∞	∞	∞	∞	One million times

Schaumont, Table 8.1 - Types of memories

la memoria distribuita complica significativamente il concetto di uno spazio centralizzato di indirizzamento della memoria, quando i dati devono essere condivisi fra componenti

- va mantenuta la consistenza di copie multiple di uno stesso dato in memorie diverse
- l'aggiornamento di un dato condiviso va realizzata in modo da non violare dipendenze di dato fra i componenti che lo condividono (v. Problema 8.1)

controllo gerarchico

una gerarchia del controllo fra i componenti assicura che l'intero SoC operi come singola entità logica

- il controllo centrale è di solito esercitato da un processore RISC, che invia comandi agli altri componenti
- questi possono operare lascamente, pressoché indipendentemente, l'uno dall'altro, tuttavia a un certo momento dovranno sincronizzarsi e inviare risultati al punto di controllo centrale

del controllo locale può essere esercitato da componenti dedicati, quali coprocessori o altro hardware custom, ma le loro operazioni e quelle del controllore centrale non sono del tutto indipendenti

per esempio, possono sincronizzarsi, sull'invio di dati per computazioni richieste e sulla ricezione di risultati da parte del controllore centrale, mediante un protocollo di handshake

il progetto di una buona gerarchia del controllo è una sfida problematica

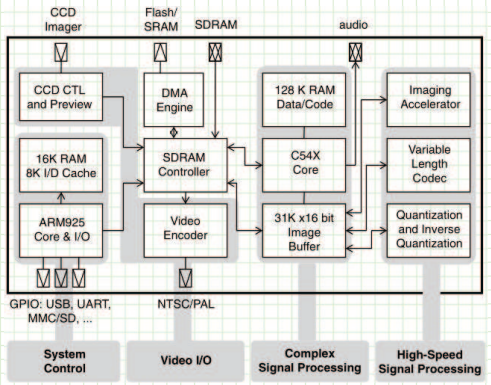
- dovrebbe sfruttare la natura distribuita del SoC il più possible
- mentre dovrebbe minimizzare i conflitti che insorgono per effetto dell'esecuzione parallela

a seconda della distribuzione del carico di lavoro, qualsiasi componente può essere un collo di bottiglia: la sfida per il progettista di SoC (o per il programmatore di una piattaforma) è di individuare tali colli di bottiglia del sistema e di controllarli

esempio: un SoC multimediale

caso di studio reale: un processore multimedia digitale da Texas Instruments

fabbricato in 130 nm CMOS, usato per l'elaborazione di immagini, audio e video in dispositivi portatili alimentati da batteria, consuma 250—400 mW



Schaumont, Figure 8.3- Block diagram of portable multi-media system

diversi modi operativi, fra cui:

- live preview di immagini (default)
- compressione video live (MPEG, MJPEG) e streaming su memoria esterna
- cattura e conversione JPEG di immagini ferme ad alta risoluzione
- cattura audio live e compressione MP3, WMA o AAC
- decodifica video e playback su schermo video di stream registrata
- decodifica e playback su schermo video di immagine ferma memorizzata
- decodifica audio e playback
- stampa di immagine memorizzata in formato adatto a stampa fotografica

quattro sottosistemi specializzati sono indicati in figura 8.3, centrati attorno al controllore SDRAM che organizza il traffico sulla grande memoria fuori dal chip, che contiene dati di immagine

le quattro proprietà discusse prima sono riconoscibili in questo chip:

- *elaborazione eterogenea e distribuita*: cablata (sottosistema video), elaborazione di segnali (DSP), general purpose (processore ARM)
- *comunicazione eterogenea e distribuita*: non vi è un bus centrale, piuttosto un controllore centrale di multiplazione degli accessi alla grande memoria fuori dal chip, a cui si aggiungono bus locali tra processori specializzati e le loro memorie
- *memoria eterogenea e distribuita*: grande SDRAM fuori dal chip, memorie per istruzioni dei processori TI DSP e ARM, memorie dati dedicate che fungono da buffer locali
- *controllo gerarchico*: una gerarchia del controllo assicura l'ottimalità del parallelismo complessivo, dove il processore ARM attiva/arresta i componenti e controlla i flussi dei dati in base al modo operativo

letture raccomandate:

Schaumont, Ch. 8, Sect. 8.1-8.3

per ulteriore consultazione:

Schaumont, Ch. 8, Sect. 8.4

Rowen, Ch. 1