

Introduzione a Electronic Design Automation

M. Favalli

Engineering Department in Ferrara

Sommario

- 1 Circuiti digitali
- 2 Flusso di progetto e ruolo della EDA

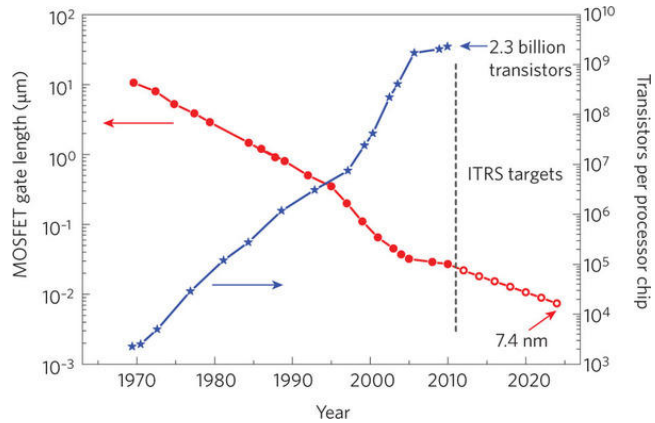
Circuiti integrati digitali

- Tecnologie
- Architetture
- Applicazioni
- Progettazione
- Produzione

Tecnologie

- Tecnologia planare del silicio
 - 3D stack
- Dispositivi MOSFET
 - FinFET
- Densità di integrazione
 - legge di Moore
 - limiti fisici
- Prestazioni
 - ritardo e consumo di potenza
 - dispositivi e connessioni
 - variabilità
- Affidabilità

Densità di integrazione



Approcci al progetto e alla fabbricazione

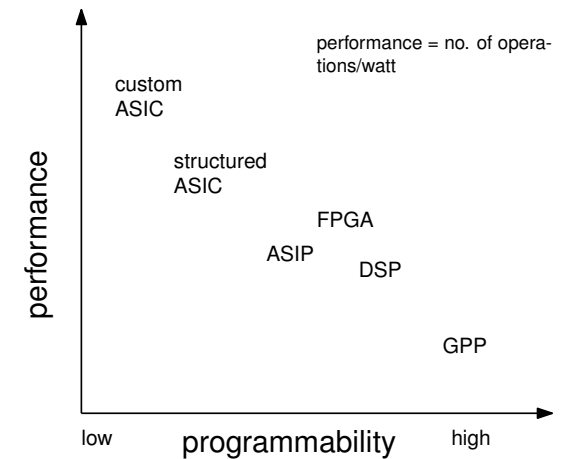
Diversi compromessi fra costi e prestazioni

- Full-custom
- Semi-custom (standard cells)
- Structured (gate-array)
- Configurable (FPGAs)

Architetture

- Processori (grandi volumi di produzione)
 - GPP, ASIP, DSP
 - full-custom
 - programmabilità
- Circuiti per applicazioni specifiche (ASIC)
 - prestazioni
 - consumo di potenza
 - input/output
 - affidabilità

Architetture



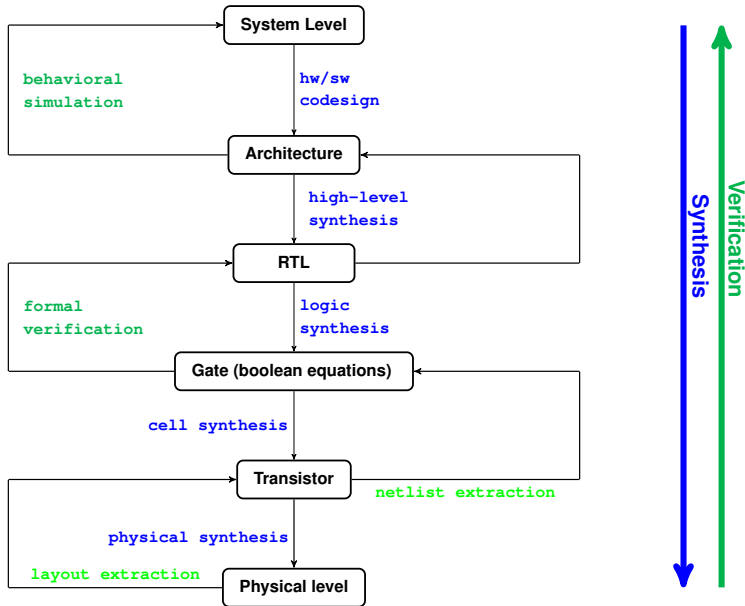
Flusso di progetto

- Approccio di tipo top-down
- Utilizzo di diversi livelli di astrazione
- Necessità di strumenti automatici di progetto
- Sintesi e ottimizzazione
- Verifica

Ruolo della design automation

- La complessità dei circuiti è tale da rendere il loro progetto non gestibile manualmente
- Praticamente ogni passo è assistito da programmi di EDA
- Il loro ruolo è fondamentale nella sintesi e nella verifica
- L'ottimizzazione di progetti sarebbe addirittura non concepibile senza di questi strumenti
- Gli algoritmi utilizzati da questi programmi provengono in parte da ambiti teorici non direttamente legati alla progettazione di sistemi digitali (logica, ricerca operativa) e in parte sono stati sviluppati specificamente per l'applicazione considerata

EDA design flow



System

- Livello di specifica
- Descrizione comportamentale
 - linguaggi GP (Java, C, C++), di modeling (Simulink), dedicati all'hardware (VHDL, SystemC, SystemVerilog) o modelli formali (algebra dei processi)
 - processi concorrenti
- Nessuna indicazione su cosa viene implementato via software e cosa via hardware
- Obiettivi di massima per costi e prestazioni

Architettura

- Struttura del sistema
- Individuazione dei principali moduli funzionali
 - una o più CPU eventualmente di vari tipi (GPU, DSP)
 - logica dedicata all'applicazione
 - memorie
 - connessioni interne e interfacce di I/O
 - componenti analogici o RF
- System-on-Chip

Sintesi: sistema \Rightarrow architettura

- Hardware-software codesign
 - analisi delle prestazioni ottenibili via sw su piattaforme off-the-shelf
 - le specifiche su banda, ritardi o consumo di potenza possono non essere soddisfacenti da questo tipo di componenti
- Individuazione dei processi o delle parti di processi computazionalmente critici che richiedono l'implementazione di logiche dedicate
- Generazione delle specifiche comportamentali per l'hardware (SystemC, Verilog, VHDL)

Verifica: sistema \Leftarrow architettura

- hw-sw co-simulation
- Operazione complessa e computazionalmente pesante
- Una volta decisa la piattaforma si conoscono le prestazioni del sw ma non quelle dell'hw che è ancora da sintetizzare
- Più raramente si usano modelli formali

Register Transfer Level - RTL

- Modella il flusso dei segnali fra registri e le operazioni logiche o aritmetiche che vengono svolte su tali segnali
- Diversi livelli di granularità:
 - sovrapposizione parziale col livello architetturale (micro-architetture)
 - livello logico comportamentale indipendente dalla tecnologia
- Diverse modalità di descrizione:
 - sequenziale: EFSM, ASM
 - concorrente
- Modello computazionale implicito: data-path e control

Sintesi: Architettura \Rightarrow RTL

- La sintesi dal livello architetturale a quello RTL può avvenire in diversi modi
- Solitamente si usa il termine high-level synthesis per descrivere le operazioni svolte
- Si possono comunque riconoscere due passaggi principali:
 - 1 l'estrazione di FSM ad alto livello dagli algoritmi contenuti nei processi
 - 2 l'estrazione di data-path e controllo da tali FSM
- Possono essere eseguite diverse ottimizzazioni
- Utilizzo di strumenti EDA per la sintesi e l'ottimizzazione

Esempio di sintesi ad alto livello

- Algoritmo che calcola il massimo comun divisore (Euclide)
- Descrizione dell'algoritmo in VHDL (C, Java, etc.)
- L'algoritmo può essere simulato o eseguito
- Pochissime informazioni sulla realizzazione hardware
- La sintesi ad alto livello come primo passo estrae un modello formale (EFSM) che corrisponde a un'implementazione sincrona

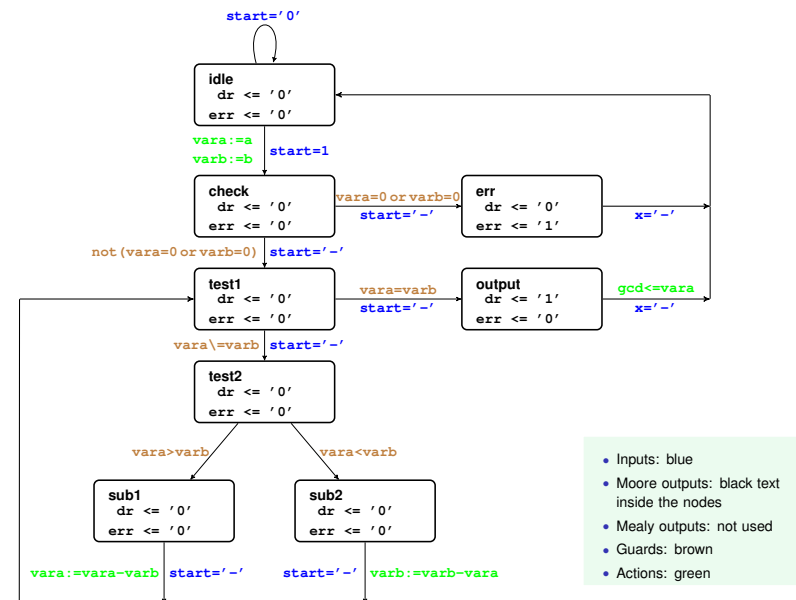
gcd - descrizione ad alto livello

```
-- high-level description of a gcd evaluator, no timing,
-- description non directly synthesizable
library IEEE;
use IEEE.STD_LOGIC_1164.all, ieee.numeric_std.all;
entity gcd is
  port (
    a : in STD_LOGIC_VECTOR(7 downto 0);
    b : in STD_LOGIC_VECTOR(7 downto 0);
    gcd : out STD_LOGIC_VECTOR(7 downto 0)
  );
end gcd;
architecture behav of gcd is
begin
```

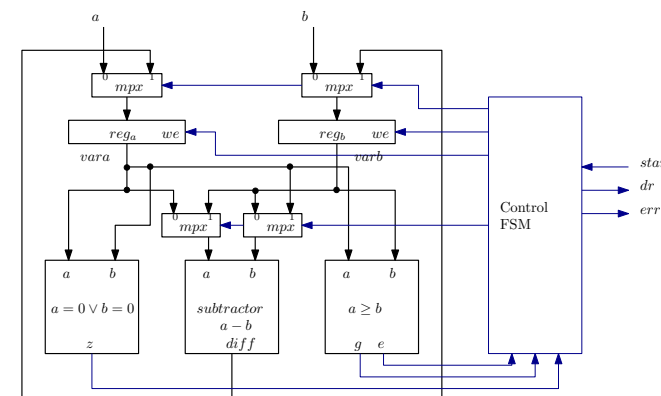
gcd - descrizione ad alto livello

```
process (a,b)
  variable vara,varb: unsigned(7 downto 0);
  constant zero: std_logic_vector(7 downto 0):=(others=>'0');
begin
  if (a=zero) or (b=zero) or (is_x(a)) or (is_x(b)) then
    gcd <= (others=>'X');
  else
    vara:=unsigned(a);
    varb:=unsigned(b);
    while (vara/=varb) loop
      if (vara<varb) then
        varb:=varb-vara;
      else
        vara:=vara-varb;
      end if;
    end loop;
    gcd <= std_logic_vector(vara);
  end if;
end process;
end architecture behav;
```

- L'algoritmo può essere trasformato in una EFSM sincrona
- Si definisce la sequenza delle operazioni da svolgere che vengono assegnate agli stati della EFSM
- Si definisce un protocollo di comunicazione con l'esterno basato su segnali di `start`, `data-ready` e di `error`
- Vengono implicitamente definiti un controllo e un data-path



- Il modello di EFSM corrisponde naturalmente al paradigma di progetto basato su data-path e controllo
- Il data-path è costituito da registri, multiplexer, blocchi logici e aritmetici
- Il controllo è una FSM convenzionale che interagisce con l'ambiente esterno alla EFSM tramite gli ingressi e le uscite della EFSM, e con il data-path tramite:
 - segnali di uscita che controllano il data-path (determinati dalle action)
 - segnali di ingresso dal data-path che forniscono le condizioni individuate dalle guard
- La sintesi è in grado di analizzare l'EFSM, estrarre un modello del data-path e uno del controllo (FSM) e produrre una descrizione al livello RTL



Analisi: Architettura \Leftarrow RTL

- Simulazione (co-simulation)
- Metodi formali (ove possibile)
- L'analisi non si limita a verificare la correttezza funzionale, ma migliora anche le previsioni sulle prestazioni del sistema digitale

Logic level: sintesi e ottimizzazione di FSM

- FSM descritta come STG
- Minimizzazione del numero degli stati
- Codifica ottimizzata dello stato
- Estrazione delle reti combinatorie che realizzano le funzioni di stato futuro e uscita

Logic level

- Punto di partenza per la sintesi
 - FSM (controllo)
 - descrizione livello RTL per il data-path
- Punto di arrivo: descrizione della rete che realizza le funzioni di controllo e data-path mediante le celle elementari messe a disposizione dalla tecnologia (gate, o celle di FPGA)
- La sintesi consiste di diversi passaggi nei quali possono essere ottimizzati area, ritardo e consumo di potenza

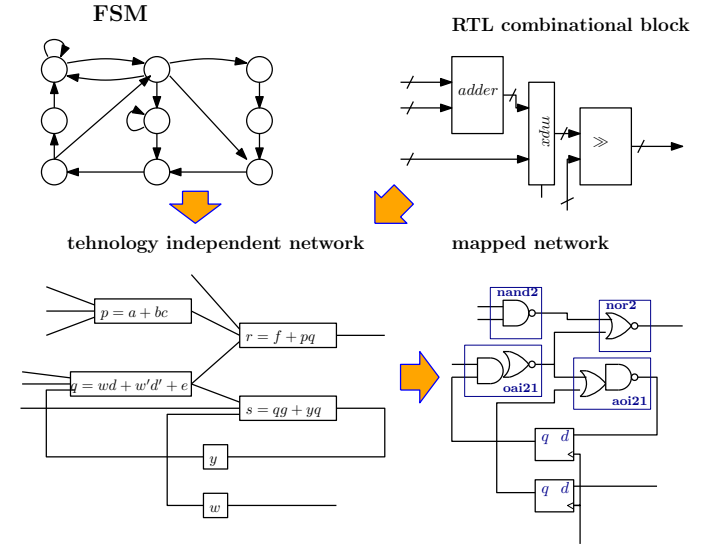
Logic level: sintesi e ottimizzazione di reti combinatorie

- Specifiche funzionali (tabella di verità, rete combinatoria)
- Trasformazione in un qualche formato che consenta la descrizione e trasformazione della rete
- Applicazione di algoritmi di sintesi e ottimizzazione che tipicamente producono una descrizione indipendente dalla tecnologia della rete (insieme di equazioni logiche)
- Obiettivi dell'ottimizzazione: area, ritardo e consumo di potenza

Logic level: technology mapping

- Mappa la descrizione indipendente dalla tecnologia sulle celle messe a disposizione dalla tecnologia stessa
- L'operazione viene fatta ottimizzando area, ritardo o consumo di potenza
- Gli algoritmi dipendono dal tipo di tecnologia (standard-cell o FPGA)

Esempio



Circuit level

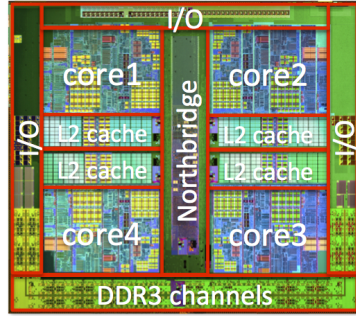
- A questo livello vengono progettate e ottimizzate le porte logiche che vengono messe a disposizione dalla tecnologia
- Transistor sizing
- Caratterizzazione delle celle dal punto di vista di ritardi e consumi

Physical level

- Il livello fisico di progetto si occupa di produrre un layout realizzabile fisicamente come circuito integrato a partire dalla struttura logica/circuitale del sistema
- A questo livello di determinano in buona parte area, prestazioni (ritardi) e consumo di potenza che ai livelli superiori erano note in maniera approssimata
- Ruolo fondamentale dell'ottimizzazione, tipicamente con algoritmi basati sulla teoria dei grafi

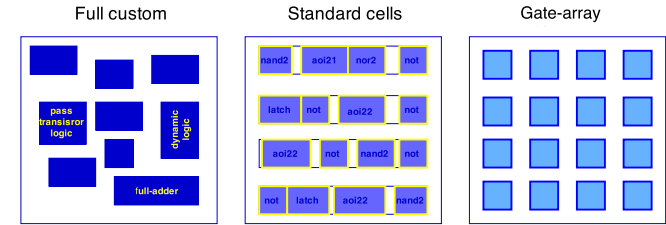
Physical level - floorplanning

- blocchi (rettangoli) fisici che contengono il layout di moduli al livello architetturale
- da ALU a intere CPU, memorie cache
- soft o hard
- ottimizzazione: area o lunghezza delle connessioni



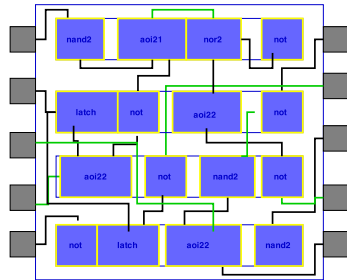
Physical level - placement

- dato un insieme di celle o macro di dimensioni fisse assegna una posizione a una cella
- ottimizzazione: area, timing o lunghezza delle connessioni
- diversi stili di progetto



Physical level - routing

- routing di cammini per i conduttori che portano i segnali del circuito
- diversi livelli di metallizzazioni
- ottimizzazione dei ritardi
- problemi specifici come il segnale di clock



Verifica di progetto

- Nonostante l'uso di strumenti automatici di progetto, "comunque necessario verificare la consistenza dell'implementazione finale con le specifiche"
- La simulazione logica ai vari livelli consentiti dai Linguaggi di descrizione dell'hardware è una soluzione parziale in quanto non consente una verifica esaustiva delle funzionalità
- I metodi formali consentono di verificare in modo esatto se una certa implementazione corrisponde a una specifica, ma possono trattare solo determinate categorie di circuiti
- La prassi industriale corrente consiste nell'utilizzo di entrambe le tecniche

Livelli nella verifica di progetto

- Al livello piú basso si ha l'estrazione dei transistori dal layout
- Da queste reti di transistori al livello switch vengono estratti i blocchi logici (gate, flip-flop) o viene eventualmente segnalata la presenza di blocchi che non rispettano alcune regole
- Dopo parte la verifica che le reti logiche ottenute siano corrispondenti al livello RTL o a quello comportamentale

Produzione e collaudo

- Una volta realizzato il layout il circuito puó essere prodotto
- I circuiti prodotti devono poi essere collaudati per garantire l'assenza di difetti di produzione
- Il collaudo necessita di sequenze in grado di rivelare l'eventuale presenza di questi difetti
- Svolgere questa operazione in circuiti con 1G di dispositivi e soltanto 400-800 pin disponibili per controllare e osservare i nodi interni non é semplice
- Sono necessari adeguati algoritmi e metodologie di progettazione che rendano il circuito facilmente collaudabile

Produzione e collaudo

- Una volta realizzato il layout il circuito puó essere prodotto
- I circuiti prodotti devono poi essere collaudati per garantire l'assenza di difetti di produzione
- Il collaudo necessita di sequenze in grado di rivelare l'eventuale presenza di questi difetti
- Svolgere questa operazione in circuiti con 1G di dispositivi e soltanto 400-800 pin disponibili per controllare e osservare i nodi interni non é semplice
- Sono necessari adeguati algoritmi per la generazione di sequenze di collaudo e metodologie di progettazione che rendano il circuito facilmente collaudabile

Generazione di vettori di collaudo

- Si tratta di generare sequenze che verificano che il dispositivo sia privo di difetti
- Questo richiede modelli di guasto che astraggano i comportamenti fisico-elettrici indotti dai difetti fisici:
 - efficienza
 - accuratezza
- Il problema della generazione di queste sequenze é computazionalmente pesante e condiziona anche la progettazione dei sistemi digitali stessi
 - problema risolto nelle reti combinatorie
 - intrattabile in molte reti sequenziali

Sistemi tolleranti ai guasti

Questi sistemi verranno brevemente discussi dal punto di vista dell'impatto dei problemi di affidabilità tipici delle tecnologie nanometriche sul collaudo e sugli strumenti di EDA in generale

Panorama industriale nella EDA

- Grandi compagnie (Synopsis, Cadence)
 - forniscono una vasta gamma di prodotti che supporta l'intero processo di progetto, verifica e collaudo (includendo anche aspetti fisici)
- Compagnie più piccole (Aldec, Mentor Graphics) che supportano alcuni aspetti (ad esempio la simulazione) con prodotti ottimizzati
- Software "in-house" sviluppato all'interno di grandi aziende di semiconduttori
- Il caso particolare degli FPGA
 - flussi di progetto low-cost
- Freeware