

Hardware/Software Co-Design

Übungsskript

Institut für Technik der Informationsverarbeitung, Karlsruher Institut für Technologie

Kapitel 1: Einleitung

Aufgabe 1.01: Hardware/Software Co-Design

Diskutieren Sie die Vor- und Nachteile der Implementierung eines Systems in Hardware bzw. Software bzgl. folgender Kriterien:

a) Entwurfszeit bzw. time-to-market

- | | | |
|-------|----|--|
| HW | -- | Lange Entwicklungszeiten für full custom ASICs und Boards; Langer externer Fabrikationsweg (z.B. für Waver); „First time right“; Hoher Testaufwand |
| HW | - | Einsatz von off-the-shelf components/module, dadurch i.a. komplizierteres Board |
| SW | ++ | Erste Lösungen sind sehr schnell fertig; Fertigstellung kann in Stufen erfolgen |
| SW | + | Softwarefehler können Post-Fabrication durch Softwareupdates beseitigt werden. |
| HW | + | Nutzung rekonfigurierbarer Hardware (FPGAs): Post-Fabrication HW Updates |
| HW/SW | + | Gleichzeitige Entwicklung möglich |

b) Performanz

- | | | |
|----|----|---|
| HW | ++ | Nichts ist schneller als ein full-custom ASIC |
| HW | + | FPGAs sind sehr schnell, vor allem wenn Parallelität ausgenutzt werden kann |
| SW | -- | Bestimmte Operationen können nur von dezidiert Hardware ausgeführt werden (z.B. restriktive Echtzeit-Bedingungen) |
| SW | + | Die Rechenleistung kann auf mehrere Prozessoren verteilt werden |
| SW | + | Optimierung durch komplexere Algorithmen, die nur teilweise effizient in HW realisierbar sind |

c) Kosten

- | | | |
|----|----|---|
| SW | ++ | Programmierkenntnisse sind weit verbreitet; Compiler sind ausgereift und leicht zu bedienen; Änderungen an der Sprache sind relativ selten; geringe Tool-Kosten |
|----|----|---|

SW	++	Off-the-shelf Mikrocontroller und Zubehör sind bedingt durch die hohen Stückzahlen billig
SW	++	Gefahr eines totalen Re-Designs ist klein
HW	--	Hohe Tool- und Personalkosten durch lange Entwicklungszeiten
HW	--	ASICs lohnen sich nur für große Stückzahlen
HW	+	Rekonfigurierbare Hardware minimiert die Entwurfskosten im Vergleich zum ASIC-Entwurf
HW	+	Rekonfigurierbare Hardware kann benötigte Chipfläche reduzieren (dyn. Rekonfiguration, Compute in „Time & Space“)

d) Leistungsverbrauch

HW	++	Hardware oft speziell auf das Design angepasst, verbraucht nur die absolut notwendige Leistung
HW	++	Nicht benötigte Teile können abgeschaltet werden
SW	--	Hohe Taktraten für die Erfüllung von Echtzeitbedingungen notwendig
SW	--	Software verbraucht in der Regel mehr Leistung als Hardware. (dynamisches Steuerwerk, Programmspeicher, Caches, etc.)
HW	-	Rekonfigurierbare Hardware vergleichsweise viel Energie. Abhilfe: Low Power Flash FPGAs (Igloo FPGA von Microsemi)

e) Wartbarkeit bzw. Änderbarkeit

HW	--	ASICs sind oft nicht-modulare Einzellösungen (=> IP Re-use)
HW	--	Ein ASIC-Layout ist fest, keine nachträgliche Änderung
SW	++	Fast beliebige Änderungen möglich
HW	+	FPGAs sind zur Laufzeit konfigurierbar (rekonfigurierbar)
HW/SW	+	Rekonfigurierbare HW ermöglicht Änderung von SW & HW

f) Testbarkeit

HW	++	CAD-Tools helfen sehr bei Entwurf und Synthese von korrekten Schaltungen
HW	+	Zusatzlogik zum Testen und Debuggen kann direkt in die Schaltung integriert werden (JTAG Interface)
HW	--	Simulation der HW sehr Zeitaufwändig und Ressourcenhungrig

SW	++	Sehr ausgereifte Entwicklungsumgebungen zum Debuggen, Testen und Simulieren vorhanden.
SW	-	Unsauberes Design wegen kurzen Deadlines möglich (=> Software-Engineering).
HW/SW	+	Rapid-Prototyping mit rekonfigurierbarer Hardware möglich / HiL (Hardware in the Loop)
HW/SW	--	Oft keine geeigneten Hilfsmittel für den Test von gemischter Hardware/Software

g) Sicherheit

HW	++	Redundanz durch Einsatz mehrerer gleichartiger Komponenten
HW	++	Hardware ist schwierig zu kopieren (Custom-ASICs, Multilayer Boards)
SW	--	Software ist relativ einfach zu kopieren oder verändern, Schutzmechanismen aufwendig oder auf Hardwaresupport angewiesen
HW/SW	+	Software passt nur zur entsprechenden Hardware
HW/SW	++	Realisierung von SIL4 (Safety Integrity Level) durch redundante Realisierung in HW und SW.

Aufgabe 1.02: Architekturen

Überlegen Sie sich Kriterien, welche die Entscheidung zur Realisierung einer Spezifikation in Hardware oder Software begünstigen.

a) Was sind die wichtigsten Kriterien im Falle einer Zielarchitektur für

- Steuerung einer Ampel:
Steuerungsdominant, Sicherheit, Einfache Konfigurierbarkeit, Umweltresistenz, Größe
- Mobiltelefon:
Steuerungs- und Datenfluss, Geringer Leistungsverbrauch, Größe, Stückzahl, zu realisierende Dienste (QoS)
- System zur Bildverarbeitung:
Datenflussdominant, Performanz, Durchsatz
- Kraftwerksüberwachung:
Steuerungsdominant, absolute Sicherheit, Redundanz, Reaktionszeit, harte Echtzeitbedingungen

b) Welche unterschiedlichen Optimierungskriterien machen die Entscheidung Hardware/Software aus im Falle einer

- Ein-Chip HW/SW-Lösung:
Kosten (nur günstig bei großen Stückzahlen), Gewicht, Größe, Zuverlässigkeit (Schirmung, Konnektoren), Leistungsverbrauch, interner vs. externer Kommunikationsaufwand, Kopierschutz
- Board-Level HW/SW-Lösung:
Erfüllbarkeit (Passt nicht auf einen Chip), Kosten (Standardchips sind günstiger), Entwurfszeit, Flexibilität, Verlässlichkeit

Geben Sie an, für welchen Anwendungs- bzw. Aufgabenbereich die eine bzw. die andere Lösungsvariante Vorteile bzw. Nachteile hat.

c) Für welche Anwendungsbereiche erscheinen Ihnen ASIPs (Prozessoren mit anwendungsspezifischem Instruktionssatz) sinnvoll? Versuchen Sie, für diese Anwendungsbereiche Charakteristika einer optimalen Architektur zu extrahieren und vergleichen Sie diese Anforderungen mit Realisierungen Ihnen bekannter ASIP-Architekturen (z.B. DSPs- Digitale Signalprozessoren).

Punkte, die für den Einsatz eines ASIPs statt einer Standard-CPU (DSP) sprechen:

- Kostengünstiger bei großen Stückzahlen, Leistungsverbrauch geringer, Operationsverkettung möglich, Parallelität möglich, spezialisierte Funktionen, Anpassung der Wortlänge, optimierte Speicherstrukturen, optimierter Datenpfad, Spezialregister.
- Aufgaben fest umrissen, keine Änderung wahrscheinlich, speziell zugeschnittener Befehlssatz erlaubt erhöhte Performanz durch Einsatz dedizierter Funktionseinheiten (heterogener vs. homogener Registersatz)