

Gliederung der Vorlesung Rechnerarchitekturen II

(Die Teile IV. bis VIII. werden je nach zur Verfügung stehender Zeit gehalten)

0. Übersicht

I. Programmierbare Schaltungen

- 1.2 Gate Arrays
- 1.3 Familien Programmierbarer Schaltungen zur Implementation von Logikfunktionen
 - 1.3.1 Programmierbare ROMs (PROMs)
 - 1.3.2 Programmable Logic Arrays (PLAs)
 - 1.3.3 Programmable Array Logic (PALs)
 - 1.3.4 CPLDs
 - 1.3.5 FPGAs
 - 1.3.5.1 CLB - Configurable Logic Block
 - 1.3.5.2 IOB - I/O Block
 - 1.3.5.3 Programmierbare Verbindungen
 - 1.3.5.4 FPGA-Familien
- 1.4 Design-Ablauf
 - 1.4.1 Spezifikation
 - 1.4.2 Entwurfsmethoden - Design Entry
 - 1.4.3 Designregeln

II. Schaltungsdesign

- 2.1 Synchrones Design
- 2.2 Vermeidung von Race Conditions
- 2.3 Erzeugung von Verzögerungen
- 2.4 Glitches
 - 2.4.1 Asynchroner MUX
 - 2.4.2 Nachsynchronisierter MUX
- 2.5 Clock Gating
- 2.6 Set-up/Hold-Time-Verletzungen
- 2.7 Metastabilität
- 2.8 undefinierte Pegel - Floating Nodes
- 2.9 Bus Contention

III. VHDL

- 1 EINFÜHRUNG
 - 1.1 HILFSMITTEL UND LITERATUR
- 2 SYNTHESE EINFACHER SCHALTWERKE
 - 2.1 ENTITY, ARCHITECTURE UND SIGNALE
 - 2.1.1 Lokale Signale
 - 2.1.2 Richtung von Entitysignalen
 - 2.1.3 Bussignale
 - 2.2 SIMULATION VON VHDL-ENTWÜRFEN
 - 2.3 SCHALTNETZE MIT BOOLE'SCHEN OPERATOREN
 - 2.4 SELEKTIVE UND BEDINGTE SIGNALZUWEISUNG
 - 2.5 VERZÖGERUNG, ZEITMODELLE
- 3 PROZESSE
 - 3.1 DEKLARATION UND AUSFÜHRUNG VON PROZESSEN
 - 3.2 SCHALTNETZE MIT SEQUENTIELLEN ANWEISUNGEN
 - 3.2.1 Case-Anweisung
 - 3.2.2 If-Anweisung (if-then-elsif/else)
 - 3.2.3 Schleifen
 - 3.2.4 Wait-Anweisung
 - 3.3 D-FLIPFLOPS, REGISTER, SCHIEBEREGISTER
 - 3.4 LATCHES
 - 3.5 KOMBINATORISCHE SCHLEIFEN
 - 3.6 SYNTHESERICHTLINIEN
 - 3.6.1 Taktsynchrone Umgebungen
 - 3.6.2 Außerhalb taktsynchroner Umgebungen

- 4 DATENTYPEN STD_ULOGIC UND STD_LOGIC
 - 4.1 TRISTATE-TREIBER
 - 4.2 KONVERSION DER DATENTYPEN BIT UND BIT_VECTOR
- 5 ARITHMETIK UND SYNCHRONZÄHLER
 - 5.1 ARITHMETIK-OPERATOREN UND ZUGEHÖRIGE DATENTYPEN
 - 5.2 SYNCHRONZÄHLER
 - 5.3 ARITHMETIK MIT DEN DATENTYPEN SIGNED UND UNSIGNED
 - 5.4 INTEGER-ARITHMETIK
 - 5.4.1 Konversionsfunktionen
- 6 ZUSTANDSAUTOMATEN
- 7 STRUKTURELLER VHDL-ENTWURF
 - 7.1 KOMPONENTENDEKLARATION
 - 7.2 KOMPONENTENINSTANTIIERUNG UND PORT MAP
 - 7.3 KONFIGURATION
 - 7.4 PORT-MODI DER VERSCHALTETEN SIGNALE IN TRUKTURBESCHREIBUNG
 - 7.5 FUNKTIONEN UND PROZEDUREN
 - 7.6 PACKAGES

IV. Speicherverwaltung bei Prozessoren

V. Cache-Speicher

- 5.1 Warum braucht man Cache-Speicher?
 - 5.1.1 Speicherhierarchie
 - 5.1.2 Anwendungen von Caches
- 5.2 Cache-Strukturen
- 5.3 Aufbau und Organisation
- 5.4 Cache-Architekturen
 - 5.4.1 Voll-assoziativer Cache
 - 5.4.2 Direct-Mapped Cache
 - 5.4.3 Mehrfach-assoziativer Cache
- 5.5 Cache-Strategien
- 5.6 Zusammenfassung

VI. Einsatz von Pipelining bei Mikroprozessoren

- 6.1 Einleitung
 - 6.1.1 Wie ist diese Steigerung möglich?
- 6.2 Ein einfaches Prozessormodell
 - 6.2.1 Befehlsphasen
 - 6.2.2 Lineare Befehlsfolge
 - 6.2.3 Beschleunigung durch Pipelining
 - 6.2.4 Pipeline-Tiefen üblicher Prozessoren
- 6.3 Strukturhasards
- 6.4 Datenhasards
- 6.5 Kontrollflußhasards
- 6.6 Zusammenfassung

VII. CISC, RISC

- 7.1 CISC – RISC
- 7.2 Hintergrund
- 7.3 Merkmale von RISC-Architekturen
- 7.4 Beispielarchitektur SPARC
- 7.5 Zusammenfassung

VIII. Beispiel-Architekturen von Prozessoren