

ONTWERP-SOFTWARE

- **Programmeerbare bouwstenen hebben een steeds toenemende complexiteit**
 - SPLD = Simple PLD (PAL en GAL)
 - CPLD = Complex PLD (EPLD)
 - FPGA = Field Programmable Gate Array
- en bevatten duizenden te programmeren punten
- **Hulp van de computer is onmisbaar om een bepaalde functie onder te brengen in een PLD**
 - hulp bij het ingeven en simuleren van het ontwerp
 - hulp bij het ‘vertalen’ van dit ontwerp naar een PLD-schakeling
 - hulp bij het configureren van de PLD
- **EDA = Electronic Design Automation**
CAEE = Computer Aided Electronic Engineering

Programmeerbare Logica

DIGITALE ELEKTRONICA 2EL

ONTWERP-SOFTWARE voor SPLD : PALASM

- **SPLD = Simple PLD (PAL + GAL)**
 - eerst op de markt gebracht door de firma MMI (Monolithic Memories Inc), die later werd opgekocht door AMD
- **PALASM (=PAL-assembler)**
 - door AMD/MMI (uitvinders v/d PAL) geleverde ontwikkelingsomgeving
 - staat heel kort bij de component (databoek is nodig!)
 - is een programmeertaal met een vrij eenvoudige syntax
 - schakeling kan op verschillende manieren worden ingegeven:
 - » logische vergelijkingen
 - » waarheidstabellen
 - » toestandmachines (FSM = Finite State Machine)
 - zorgt zelf voor minimalisatie
 - biedt de mogelijkheid tot simulatie
 - genereert de JEDEC-file (= fuse-map) die het programmeertoestel nodig heeft om de PLD te programmeren

Programmeerbare Logica

DIGITALE ELEKTRONICA 2EL

ONTWERP-SOFTWARE voor SPLD : ABEL

- **ABEL = Advanced Boolean Expression Language**
 - op de markt gebracht door DATA-I/O, een fabrikant van programmeer-toestellen (en dus niet van PLD's zelf)
 - staat op een hoger niveau (*compiler*) dan PALASM (*assembler*)
 - » heeft complexere syntax-constructies
 - kent bv. de instructie `COUNT := COUNT + 1` om een teller te beschrijven
 - bij PALASM moet je zelf de vergelijkingen voor de verschillende flipflops afleiden
 - » kan meer onafhankelijk van de component gebruikt worden
 - moet uiteraard hetzelfde doel vervullen
 - » *design entry* (met *equations*, *truth tables* en/of *state diagrams*)
 - » *design simulation*
 - » *logic synthesis* (= vertalen naar JEDEC-file)
 - ook nu nog vaak onderdeel van grotere ontwerp-omgeving
- **Andere PLD-compiler : CUPL (van Logical Devices)**

Programmeerbare Logica
DIGITALE ELEKTRONICA 2EL

ONTWERP-SOFTWARE voor CPLD en FPGA

- **Complexere componenten vragen ook een ontwerp-omgeving met een hogere complexiteit**
 - we gaan een FPGA met 10000 gates niet beschrijven met alleen maar Booleaanse vergelijkingen
- **Ontwikkelomgeving =**
 - FRONT-END**
 - staat het kortst bij de gebruiker
 - bevat DESIGN ENTRY en SIMULATION
 - vaak van een onafhankelijke software-fabrikant
 - BACK-END**
 - staat het dichtst bij de component
 - bevat DESIGN IMPLEMENTATION
 - altijd van de fabrikant van de gekozen PLD

Programmeerbare Logica
DIGITALE ELEKTRONICA 2EL

DESIGN ENTRY

Twee mogelijkheden voor DESIGN ENTRY :

- **Schematic Capture**

- tekenpakket om op grafische wijze het schema van de schakeling in te voeren
- wordt onoverzichtelijk voor grote ontwerpen

- **HDL = Hardware Description Language**

- de schakeling wordt 'beschreven' met een soort programmeertaal
 - » beschrijving van *de structuur* en/of *het gedrag* van de schakeling
 - » vergelijk met ABEL, maar op een hoger niveau
 - » oorspronkelijk bedoeld voor specificatie en documentatie
 - » geschikt voor simulatie en voor automatische synthese
- in Europa : VHDL = VHSIC Hardware Description Language
(VHSIC = Very High Speed Integrated Circuit)
- in Amerika : VERILOG

Programmeerbare Logica

DIGITALE ELEKTRONICA 2EL

SIMULATIE

Twee mogelijkheden voor SIMULATIE :

- **functionele simulatie**

- gebeurt onmiddellijk na het ingeven van het schema (of na synthese van de HDL-code)
- controleert de logische werking van de schakeling
- houdt geen rekening met vertragingen of hanteert 'unit-delay'

- **timing simulatie**

- kan pas gebeuren nadat de schakeling volledig is geïmplementeerd in de gekozen component
- op dat ogenblik kent de software de juiste vertragingen
- er gebeurt nu een 'back-annotation', waarna de (zelfde) functionele simulatie opnieuw kan gebeuren, maar nu wel rekening houdende met de exacte te verwachten vertragingen

Programmeerbare Logica

DIGITALE ELEKTRONICA 2EL