Digitale Techniek

Een inleiding in het ontwerpen van digitale systemen

Jesse op den Brouw

Eerste druk

Digitale Techniek

Een inleiding in het ontwerpen van digitale systemen

Jesse op den Brouw

De Haagse Hogeschool

Eerste druk

Eerste druk

www.delftacademicpress.nl/e031.php

ISBN: 97890-6562-4468
NUR: 173/959

De auteur schenkt de royalties aan Stichting KiKa.

De foto op de omslag is gemaakt door Gerald van Kampen.

Suggesties en/of opmerkingen over dit boek kunnen worden gestuurd naar:
J.E.J.opdenBrouw@hhs.nl.

Voorwoord

Dit is een boek over digitale techniek. De reden waarom ik dit boek geschreven heb is dat ik de bestaande boeken niet toereikend vind. Er zijn drie beweegredenen voor de realisatie van dit boek: de onderwerpen, de Nederlandse taal en de prijs.

Een van de eigenschappen van docenten is dat ze het materiaal van anderen nooit perfect vinden voor hun eigen lessen. Slides worden toch nét weer op een andere volgorde gezet dan het boek suggereert en sommige hoofdstukken uit het voorgeschreven boek worden overgeslagen. Daarnaast missen veel boeken nog wel eens stof die wel behandeld zou moeten worden.

Er zijn nauwelijks Nederlandse boeken te vinden op het gebied van digitale techniek. Een eenvoudige zoekopdracht op een bekende boeken-site levert een tiental boeken op waarvan de meest recente in 2008 is verschenen. Er zijn zeker goede Engelse boeken te vinden over digitale techniek. Het nadeel is dat deze boeken over het algemeen vrij duur zijn, dat een drempel vormt voor studenten om het boek aan te schaffen. Ook de taal is hieraan debet.

Dit boek is geschreven voor studenten van de Nederlandstalige hbo-opleiding Elektrotechniek. De populatie wordt gevormd door een mengelmoes van niveaus: havo-ers, mbo-ers, vwo-ers, overstappers van andere hbo-opleidingen en tu-opleidingen. Dat levert nog wel eens problemen op met het niveau van de lessen: sommige studenten vinden de onderwerpen te makkelijk en anderen juist te moeilijk. Dit boek probeert een balans te vinden tussen de twee uitersten. Er zijn extra paragrafen opgenomen met verbredende en verdiepende informatie, zoals vereenvoudigen met de Quine-McCluskey-methode. Voor sommigen is bepaalde stof toch nog moeilijk te begrijpen. Er is getracht om alle principes en voorbeelden goed en uitvoerig te beschrijven. Sommige studenten hebben echter al aan een half woord genoeg.

Wat betreft de Nederlandse taal is dit boek in twee schrijfstijlen geschreven. Enerzijds wordt er informeel geschreven, andere stukken zijn wat formeler opgemaakt. Dat geeft de lezer een prettige afwisseling. Waar nodig zijn Engelse termen gebruikt omdat de industrie die nou eenmaal gebruikt.

Dit boek is opgemaakt in LaTeX [1] (LaTeX-engine = pdfLaTeX 1.40.21). LaTeX leent zich uitstekend voor het opmaken van lopende tekst, tabellen, figuren, programmacode, vergelijkingen en referenties. De gebruikte LaTeX-distributie is TexLive uit 2019 [2]. Als editor is TexStudio [3] gebruikt. Tekst is gezet in Charter [4], een van de standaard

fonts in LaTeX. De keuze hiervoor is dat het een prettig te lezen lettertype is, een *slanted* letterserie heeft en een bijbehorende wiskundige tekenset heeft. Code is opgemaakt in Nimbus Mono [5] met behulp van de *listings*-package [6]. Karnaughdiagrammen zijn opgemaakt met de *askmaps*-package, door de auteur ontwikkeld en beschikbaar gesteld op CTAN [7]. Voor het tekenen van toestandsdiagrammen is *TikZ/PGF* gebruikt [8].

Alle figuren, foto's en broncodes zijn door de auteur zelf ontwikkeld.

Leeswijzer

Hoofdstuk 1 geeft een algemeen overzicht van de digitale techniek. Het begint met een uiteenzetting waarom digitale techniek zo'n belangrijke plaats in de elektrotechniek heeft gekregen. Daarna volgt een lijst met belangrijke gebeurtenissen in de digitale techniek. We kijken kort naar het digitaliseren van analoge signalen om vervolgens te laten zien dat het ontwerpen van digitale systemen op een ordelijke manier moet gebeuren door middel van een ontwerptraject. We bespreken digitale bouwstenen aan de hand van elektrische eigenschappen, logische werking en tijdgedrag. De logische werking wordt ondersteund door het gebruik van waarheidstabellen, logische functie en schemasymbolen. Om snel enkele regels van de schakelalgebra te begrijpen sluiten we af met een eerste blik op logische schakelingen.

In hoofdstuk 2 worden de decimale, binaire en hexadecimale talstelsels uitgelegd. Bijna alle digitale systemen gebruiken het binaire talstelsel om informatie te verwerken. Het octale talstelsel wordt kort toegelicht. Er wordt een introductie gegeven op fracties (het deel van een getal na de komma). Vervolgens wordt de conversie tussen de talstelsels besproken. We gaan hierbij steeds uit van getallen groter dan of gelijk aan 0. We besteden ook aandacht aan het bepalen van het aantal bits voor een decimaal geheel getal en laten enige voorkomende bitbreedtes zien. We bespreken de BCD-code die soms gebruikt wordt bij numerieke gegevens. Als laatste besteden we aandacht aan een aantal coderingen, onder andere voor karakters.

Hoofdstuk 3 legt het fundament voor de schakelalgebra, de wiskunde achter de digitale schakelingen. Rekenregels worden uitgelegd en de som-van-producten-vorm en product-van-sommen-vorm van logische functies worden toegelicht. We geven ook enkele voorbeelden van bewerkingen met schakelfuncties. Als laatste lichten we nog het begrip don't care toe.

In hoofdstuk 4 worden combinatorische schakelingen behandeld. Er wordt getoond hoe een schakeling geanalyseerd en gesynthetiseerd (opgebouwd) wordt vanuit een schakelfunctie. We laten een aantal manieren zien om functies te vereenvoudigen waaronder schakelalgebra en Karnaughdiagrammen. Vervolgens wordt de realisatie met poorten, multiplexers en ROM's besproken. We werken in een aantal voorbeelden uit hoe een schakeling wordt gerealiseerd vanuit een geschreven specificatie. Er wordt stilgestaan bij de elektrische eigenschappen van ingangen en uitgangen zoals Schmitt-trigger, tri-state en open drain. Ten slotte lichten we het verschijnsel timing hazards toe.

Hoofdstuk 5 behandelt de vier elementaire binaire rekenoperaties: optellen, aftrekken, vermenigvuldigen en delen. Het eerste deel van dit hoofdstuk toont de bewerkingen voor natuurlijke getallen, gehele getallen groter dan of gelijk aan 0. We laten zien hoe de op-

telling, aftrekking en vermenigvuldiging kunnen worden gerealiseerd met schakelingen. Het tweede deel introduceert het gebruik van gehele getallen met teken. De bekende wijze met plus- en minteken is niet geschikt voor het realiseren van eenvoudige schakelingen. We bespreken de two's complement representatie die wel leidt tot realisatie van eenvoudige schakelingen. Een aantal belangrijke aspecten passeert de revue zoals bereik van getallen en overflow. We eindigen het hoofdstuk met vergelijkschakelingen en hoe het optellen en aftrekken in een microprocessor wordt gerealiseerd.

Hoofdstuk 6 gaat geheel over geheugenelementen. We behandelen diverse varianten van latches om vervolgens tot de conclusie te komen dat deze schakelingen nogal wat problemen opleveren. We laten zien dat met flipflops deze problemen grotendeels kunnen worden vermeden. Een belangrijk aspect is de timing van latches en flipflops. Kennis van dit onderwerp is belangrijk om inzicht te krijgen in het tijdgedrag van geheugenelementen en voor het ontwerpen van robuuste schakelingen. We bespreken de asynchrone set en reset waarmee flipflops tijdens het opstarten van de schakeling in een bekende stand kunnen worden gedwongen. Tot slot behandelen we een drietal voorbeelden van flipflops: een flipflop met enable (stuursignaal om de flipflop nieuwe data te laten overnemen), het register en het schuifregister.

In hoofdstuk 7 wordt de beschrijvingstaal VHDL geïntroduceerd. We beginnen met de concepten van VHDL, hoe de taal is opgebouwd en hoe het gebruikt moet worden. VHDL is nauw verwant met simulatie zodat een flink deel is ingeruimd om dit te behandelen. VHDL kan gebruikt worden voor synthese, het automatisch genereren van hardware. We vertellen wat wel en wat niet kan worden gesynthetiseerd. VHDL is een omvangrijke taal met veel taalconstructies en mogelijkheden. Daarom behandelen we de taal niet uitgebreid. Referenties naar diverse literatuur worden gegeven.

Hoofdstuk 8 staat in het teken van schuifregisters en tellers. Dat zijn twee veel voorkomende digitale bouwstenen. Er wordt uitgelegd wat een schuifregister is en wat de mogelijkheden zijn zoals linksom en rechtsom schuiven, laden en data vasthouden. Schuiven heeft ook een rekenkundige werking, namelijk vermenigvuldigen van en delen door 2. Dat is voor unsigned en signed getallen verschillend. De opbouw en VHDL-beschrijving van schuifregisters worden uitgelegd en er wordt een voorbeeld gegeven van een toepassing: de RS-232 interface. Een andere veel gebruikte component is de teller. De basis van tellen in het binaire talstelsel wordt behandeld en leidt tot de constructie van tellers op volgens het binaire telproces. In eerste instantie worden tellers die tellen met een macht van 2 behandeld. Later passeren ook andere telcycli de revue. We laten twee voorbeelden zien waarbij tellers gebruikt worden: een digitale signaalvormgenerator en een PWM-generator.

In hoofdstuk 9 gaan we dieper in op timing in kloksynchrone systemen. We bekijken directe en indirecte dataoverdracht tussen flipflops en laten zien dat het mogelijk is om de maximale toelaatbare klokfrequentie te berekenen. De timing van externe ingangen en uitgangen worden ook besproken, evenals de timing bij dataoverdracht tussen flipflops met verschillende timingparameters. Een flink deel van dit hoofdstuk is ingevuld met voorbeelden.

Hoofdstuk 10 behandelt drie onderwerpen waar we met praktische systemen mee te

maken krijgen: synchronisatie, metastabiliteit en reset. We laten eerst zien dat synchronisatie van externe ingangssignalen noodzakelijk is om de betrouwbaarheid van een digitaal system te waarborgen. Synchronisatie van uitgangssignalen kan noodzakelijk zijn om verschijnselen als hazards te voorkomen. Een reset behoort tot een vast onderdeel van elk digitaal systeem met geheugen. Het dwingt de geheugenelementen in een bekende begintoestand zodat de een correcte werking verzekerd is.

Hoofdstuk 11 staat geheel in het teken van toestandsmachines. Er wordt begonnen met een algemene introductie op dit gebied waarna de Mealy- en Moore-modellen worden besproken. Een eerste stap van het ontwerpen een toestandsmachine is het opstellen van een toestandsdiagram. Dit is de lastigste stap omdat vanuit een (vaak) geschreven specificatie een machine moet worden ontworpen. Alle volgende stappen kunnen routinematig door de ontwerper of softwaretools worden uitgevoerd. Het is voor een ontwerper goed om te weten hoe de machine in hardware wordt gerealiseerd. We bespreken verder hoe een reset moet worden geïmplementeerd en dat ingangssignalen moeten worden gesynchroniseerd. De taal VHDL wordt gebruikt om toestandsmachines direct vanuit het toestandsdiagram te beschrijven. Voorts bespreken we een tweetal typische toestandsmachines: patroonherkenners die in een seriële ingangsstroom een bepaald patroon herkennen en muntenherkenners die een bepaalde hoeveelheid ingeworpen munten herkennen. We bespreken ook toestandscoderingen en toestandsminimalisatie, hoewel dit laatste niet uitgebreid aan bod komt. Als laatste bepreken we het tijdgedrag van toestandsmachines.

In hoofdstuk 12 komen datapadsystemen aan bod. Dit is een beproefde methode om complexe digitale systemen te verdelen in kleinere onderdelen. Elk onderdeel kan volgens het datapad-besturingsmodel worden gesplitst in een datapad waarlangs de data wordt getransporteerd en gemanipuleerd en een besturing die het datapad aanstuurt zodat het geheel de juiste werking krijgt. De besturing wordt gerealiseerd middels een toestandsmachine. In VHDL is het mogelijk om deze splitsing door te voeren in de code, maar het is ook mogelijk om de acties op het datapad te integreren in de code voor de besturing.

Hoofdstuk 13 gaat over de ontwikkeling van een eenvoudige processor. De processor bestaat uit een datapad, vier registers, een teller en een besturings-ROM. Het datapad kan alleen 8-bits unsigned getallen verwerken. Het is mogelijk om de register te laden met een waarde, rekenkundige en logische bewerkingen uit te voeren en om te springen. Dit laatste kan ook op een conditie afhankelijk van het resultaat van een bewerking. We behandelen zowel hardware- als softwareaspecten. De processor wordt volledig in VHDL beschreven. Daarna laten we zien hoe we software kunnen ontwikkelen. Er wordt een voorbeeld gegeven hoe een looplicht kan worden ontwikkeld. Als laatste behandelen we enkele uitbreidingsmogelijkheden.

Studiewijzer

In onderstaande tabel wordt een overzicht gegeven van de stof die een student bij een eerste introductie onderwezen zou moeten krijgen. Uiteraard is iedereen vrij om zelf de onderwerpen te kiezen.

Deel 1

Hoofdstuk 1	1.1–1.6, 1.7.1, 1.8–1.12
Hoofdstuk 2	2.1–2.11, 2.13, 2.15
Hoofdstuk 3	3.1–3.6, 3.9, 3.10
Hoofdstuk 4	4.1, 4.2, 4.3.1, 4.3.2, 4.4–4.6, 4.8–4.11
Hoofdstuk 5	5.1–5.3, 5.6–5.14, 5.16, 5.20
Hoofdstuk 6	6.1–6.3, 6.5, 6.6, 6.8–6.13, 6.16–6.17

Deel 2

Hoofdstuk 7	helemaal
Hoofdstuk 8	8.1, 8.4, 8.6 (aanbevolen 8.2, 8.3 en 8.5)
Hoofdstuk 9	helemaal
Hoofdstuk 10	helemaal

Deel 3

Hoofdstuk 11	helemaal
Hoofdstuk 12	helemaal
Hoofdstuk 13	helemaal

Verantwoording inhoud

Dit boek voldoet aan acht van de negen punten die opgesomd zijn bij het aandachtsgebied Digitale techniek in de basis Body of Knowlegde and Skills (BoKS) Elektrotechniek zoals is vastgelegd door de HBO Engineering. Alleen het onderdeel AD/DA ontbreekt. De BoKS is te vinden via [9].

De opzet van het boek is redelijk klassiek. In het eerste deel wordt nog geen gebruik gemaakt van VHDL om de student niet af te leiden van het doel: leren wat digitale techniek inhoudt. Elk hoofdstuk wordt begeleid door een aantal vraagstukken. Veel van deze vraagstukken geven extra informatie omtrent een bepaald onderwerp. Het maken van de vraagstukken wordt daarom ook aanbevolen. In dit boek wordt de TTL-technologie slechts kort beschreven. Reden hiervoor is dat de meeste digitale schakelingen niet meer met deze technologie gemaakt worden. CMOS wordt veruit het meeste gebruikt. Daarom wordt aan deze technologie meer aandacht besteed.

Het eerste hoofdstuk geeft in vogelvlucht de digitale techniek weer. Er wordt nog niet gesproken over binaire getallen. Dit is bewust gedaan om naast de theoretische kennis ook snel aan de slag te gaan met praktische vaardigheden. Er is een paragraaf gewijd aan elektrische schakelingen voor ingangen en uitgangen omdat de studenten tijdens praktische werkzaamheden hier mee te maken krijgen.

Moderne computersystemen rekenen (voor de gebruiker althans) vrijwel allemaal in het binaire talstelsel. Een gedegen kennis hiervan is essentieel. Voor grotere getallen wordt het hexadecimale stelsel gebruikt, zeker als de student later met microcontrollers en computerarchitectuur aan de slag gaat. Getallen kunnen ook codes voorstellen. We laten de BCD-code zien die in veel systemen nog gebruikt wordt als decimale uitvoer noodzakelijk is. Verder passeert de ASCII-code de revue. Veel microcontrollers zijn uitgevoerd met een seriële interface waarover deze code wordt getransporteerd en veel

computerbestanden zijn in de ASCII-code opgeslagen.

Schakelalgebra is veelal een taaie kost. Toch is het noodzakelijk hier de grondslagen van te beheersen om een digitaal systeem te doorgronden. Combinatorische schakelingen vormen het begin van de “echte” digitale techniek. Hoewel talen als VHDL veel maskeren (denk aan het niet zichtbaar zijn van minimalisatie tijdens synthese) is het nodig om te spelen met logische bouwstenen.

Over de zin en onzin van (handmatig) minimaliseren kan gediscussieerd worden. De meeste ontwerpsoftware is al met een *minimizer* uitgerust. Minimaliseren wordt in dit boek besproken om de student er bewust van te maken dat het mogelijk is om een schakeling op meerdere manieren te realiseren en dat een bepaalde ontwerpkeuze kan leiden tot meer of minder logica. Het gebruik van talen als VHDL verbergt dat in grote mate. Het minimaliseren geeft de latere beroepsbeoefenaar *awareness* dat zulke talen hardware genereren en geen software.

Vroeg of laat moeten studenten zich ook bezig houden met de elektrische kant van een schakeling. Het is van belang om te weten hoe ingangen moeten worden aangestuurd en hoe uitgangen de rest van de schakeling beïnvloeden. Er wordt alleen gekeken naar CMOS-technologie omdat de meeste grote chips in deze technologie worden ontworpen. De TTL-technologie is grotendeels verdwenen. Soms zijn ingangen en uitgangen *TTL-compatible* zodat ze zouden kunnen worden aangesloten op TTL-(compatible) chips. De student moet zich dit deel dan zelf eigen maken. Referenties naar literatuur worden gegeven.

De two’s complement representatie is dominant in de digitale techniek en software. Er is dan ook een behoorlijk stuk ingeruimd om dit onderwerp te bespreken. De student heeft hier profijt van omdat veel andere vakgebieden, zoals microcontrollers en programmeren, gebruik maken van de two’s complement representatie.

In het hoofdstuk over geheugenschakelingen zijn de bekende schakelingen als SR-latch, gated SR- en D-latch en D-flipflop prominent aanwezig. De JK-flipflop is als verdiepend aangemerkt omdat alle moderne chips uitsluitend met D-flipflops worden uitgerust. De SR-flipflop wordt niet behandeld. Een enable-faciliteit mag alleen synchroon worden ingebouwd en er wordt toegelicht waarom clock gating beter vermeden kan worden.

De taal VHDL wordt pas na ongeveer de helft van het boek geïntroduceerd. Dat heeft als reden de student in het eerste gedeelte niet te vermoeien met allerlei zaken als opbouw van de taal, simulatie en synthese. Een te snelle introductie leidt tot onvoldoende kennis van de synthese van digitale schakelingen. Er wordt slechts een deel van de taal besproken en er wordt ingegaan op simulatie en synthese. Onderwerpen als files, pointer en records worden niet besproken.

Twee belangrijke bouwstenen in digitale systemen zijn schuifregisters en tellers. In feite zijn het toestandsmachines maar dan met een specifieke, generieke functie. We laten zien dat het eenvoudig is om schuifregisters en tellers met VHDL te beschrijven.

Timing bij dataoverdracht wordt in de meeste boeken over digitale techniek niet uitgebreid behandeld terwijl het een belangrijk onderwerp is. Een schakeling kan misschien op logisch gedrag correct werken maar niet qua timing. Alleen de timing bij kloksyn-

chrone systemen wordt behandeld omdat verreweg de meeste systemen kloksynchroon ontworpen worden.

Bij praktische systemen krijgen de studenten te maken met begrippen als synchronisatie, metastabiliteit en reset-implementatie. Het is dus vooral een praktische exercitie waarbij de begrippen worden uitgelegd.

Kennis en kunde op het gebied van toestandsmachines is essentieel voor de digitale ontwerper. Complexe systemen bestaan namelijk uit meerdere van deze machines. In feite kunnen we zeggen dat elke sequentiële schakeling een toestandsmachine is; flip-flops, registers en schuifregisters vallen hier ook onder. De taal VHDL zorgt ervoor dat de ontwerper zich niet druk hoeft te maken over allerlei routinematige klussen zoals toestandscodering en uitwerken van de schakelfuncties en alleen maar hoeft te zorgen dat de machine correct is beschreven, het liefst met een minimum aan toestanden.

In dit boek worden alleen de kloksynchrone sequentiële machines behandeld. De reden hiervoor is dat asynchrone machines in de industrie nauwelijks ontworpen worden. De overgrote meerderheid is kloksynchroon.

Processoren horen tot het arsenaal aan mogelijkheden van de ontwerper van digitale systemen. Veel systemen zijn uitgerust met een eenvoudige processor (met software) geflankeerd door speciale hardware. Te denken valt aan een systeem met een *processor core* met gespecialiseerde hardware voor het coderen van beeld en geluid. We behandelen slechts een eenvoudig ontwerp, net genoeg om de highlights van een processor te doorgronden.

Website

Op de website https://ds.opdenbrouw.nl zijn slides, practicumopdrachten en aanvullende informatie te vinden. De laatste versie van dit boek wordt hierop gepubliceerd. Er zijn ook voorbeeldprojecten voor de Quartus-software van Altera te vinden. De projecten kunnen vaak zonder aanpassingen op het DE0-bordje van Terasic uitgeprobeerd worden.

Dankbetuigingen

Dit boek had niet tot stand kunnen komen zonder hulp van een aantal mensen. Ik wil collega Harry Broeders van Hogeschool Rotterdam bedanken voor zijn geweldige bijdrage. Niet alleen op technisch gebied, maar ook taalkundig en op de indeling van dit boek. Collega Ben Kuiper heeft een prima bijdrage geleverd op technisch en taalkundig gebied. Collega Mehmet Can wordt bedankt voor zijn bijdrage op het gebied van MOS-transistoren in hoofdstuk 4.

Verder worden mijn studenten bedankt, in het bijzonder Marcel Jongkind, Frederick Kramer, Ruben de Graaff, Bob Swinkels, Daniël Vermeulen en Abraar Muhammad, voor het opsporen van enkele fouten en het geven van suggesties.

De ASCII-tabel op pagina 58 is ontwikkeld door Victor Eijkhout en is met zijn toestemming gebruikt. Deze tabel kan gevonden worden op http://www.ctan.org/tex-archive/info/ascii-chart.

De uitspraak “fenomenologische benadering” op pagina 117 is bedacht door Annel van Houts en Jan van Yperen.

Jesse op den Brouw, augustus 2020.

Inhoudsopgave