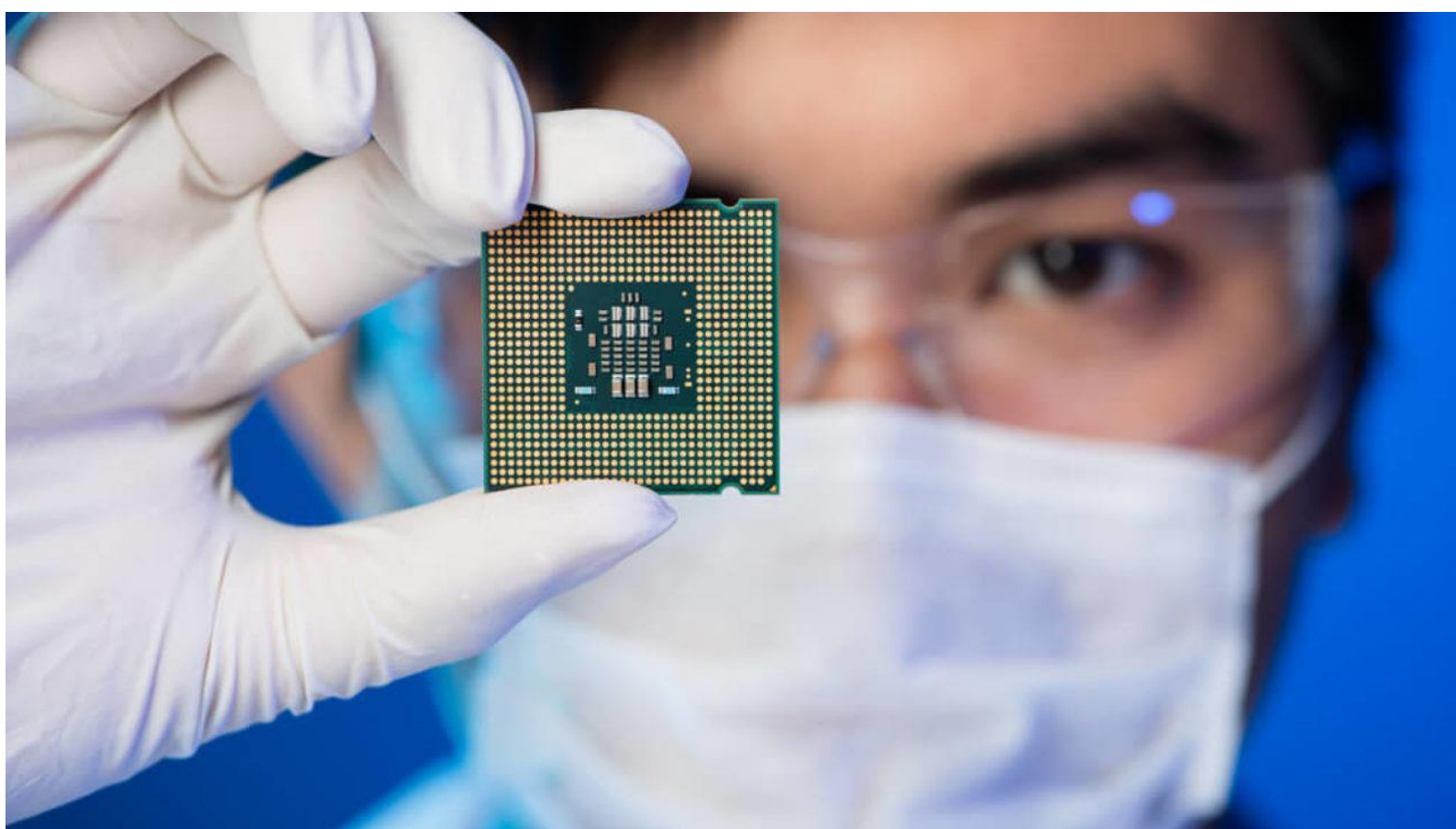


# System on a Chip (SOC)

Les handleiding



**Remko Welling**

**HAN\_**UNIVERSITY OF APPLIED  
SCIENCES

Arnhem, 23 december 2024

## Document historie

Version	Date	Reviewer	Note/Changes
001	23-12-2024	WLGRW	Start van document voor academisch jaar 2024-2025

## Todo

Date	What	Impact	Status
23-01-2019	Copyright statement shall be verified to meet HAN regulations.	Low	Pending

## Contact information

Ing. Remko Welling  
T/M: +31 (0)6 103 567 21  
E: remko.welling@han.nl

## Over dit document

In dit document wordt gebruik gemaakt van pictogrammen om extra informatie aan te geven en de prioriteit daarvan. De gebruikte pictogrammen zijn:



Opgelet: Belangrijke informatie over het onderwerp die veel invloed kan hebben.



Leerdoelstelling of leeruitkomst (LUK): Wat ga je hier leren.



Aanvullende informatie over het onderwerp.



Verder lezen: Er is meer informatie beschikbaar over dit onderwerp.



Suggestie: Een suggestie of tip om te gebruiken bij het uitvoeren van een activiteit.

## Responsible disclosure

Studenten en leraren worden aangemoedigd om ervaringen, problemen, fouten of suggesties te delen met de auteur van dit document. Stuur alstublieft jouw feedback naar de auteur van dit document zodat we het kunnen aanpassen en verbeteren. Dankjewel.

## Copyright

Dit document is onderdeel van de SOC-lessen aan de Academie Engineering en Automotive van de Hogeschool van Arnhem en Nijmegen (HAN) te Arnhem.

This paper is free: You may redistribute it and/or modify it under the terms of a Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>) by Remko Welling (<https://ese.han.nl/~rwellings/>) E-mail: [remko.welling@han.nl](mailto:remko.welling@han.nl)



All copyrights and Trademarks belong to their respective owners, Icons are not copyrighted, Cover picture source: E-Spin <https://www.e-spin.com/system-on-chip-soc-from-evolutionary-to-revolutionary/>

## Disclaimer

De auteur van dit document behoudt zich het recht voor om op elk moment wijzigingen aan te brengen zonder voorafgaande kennisgeving. Als er wijzigingen worden aangebracht, wordt het herziene document onmiddellijk gepubliceerd. Controleer het revisie overzicht om uzelf op de hoogte te stellen van eventuele wijzigingen.

## Contents

1	Les informatie. ....	5
1.1	Overzicht .....	5
1.2	Leerdoelstellingen.....	5
1.3	Lesaanpak.....	6
1.3.1	Flipped Classroom .....	6
1.4	Digitaal lesgeven .....	8
1.4.1	Online lessen .....	8
1.4.2	Opmnames .....	8
1.5	Lesmaterialen.....	8
1.5.1	Documentatie .....	9
1.5.2	Hardware.....	9
1.5.3	Software.....	11
1.6	Les planning .....	11
1.6.1	Theorielessen .....	12
1.6.2	Opdrachten .....	12
1.7	Tentamen en beoordeling.....	16
1.7.1	Sprint 1 .....	16
1.7.2	Sprint 2 .....	17
1.7.3	Sprint 3 .....	17
1.7.4	Indienen van opdrachten.....	18
1.7.5	Deadlines.....	18
1.7.6	Herkansingen .....	19
	Bibliography .....	20
	Appendix .....	21
	Vertaaltabel Pedroni 2 <sup>nd</sup> edition naar 3 <sup>rd</sup> edition .....	21

## 1 Les informatie.

Dit hoofdstuk presenteert de algemene cursusinformatie voor de System on Chip (SOC) lessen binnen de hoofdmodule Embedded Hardware Engineering (H-EHE).



De onderwijseenheid (OWE) System on Chip (SOC) is deel van de Hoofdmodule Embedded Hardware Engineering (HM EHE System on Chip) die wordt gegeven aan de deeltijdopleiding Embedded Systems Engineering (ESE).

### 1.1 Overzicht

De lessen SOC zijn een inleiding tot de hardware beschrijving taal VHDL waarmee logische schakelingen worden gerealiseerd in programmeerbare hardware zoals Field Programmable Gate Arrays (FPGA's).

Ter voorbereiding op de lessen over VHDL wordt een inleiding tot logische schakelingen gegeven. Hierin worden de basisprincipes, analyse en ontwerp van zowel combinatorische logische schakelingen en sequentiële logische schakelingen behandeld.



De lessen worden de basisprincipes van digitale techniek aangeleerd door een introductie in verschillende talstelsels die in de digitale techniek worden gebruikt. Er wordt geleerd om getallen in de verschillende talstelsels te converteren van en naar elkaar. Hoe wordt omgegaan met positieve en negatieve getallen en op welke wijze kunnen rekenkundige bewerkingen worden uitgevoerd.

Bij de behandeling van de basis van digitale bouwstenen, logische poorten, wordt booleaanse logica geïntroduceerd en de wijze waarop deze formules kunnen worden geminimaliseerd. Voor de optimalisatie van booleaanse logica wordt gebruikgemaakt waarheidstabellen en Karnaugh diagrammen.

Afsluitend worden de flipflop en haar varianten gepresenteerd met de introductie in sequentiële logica en tijddiagrammen.

In de lessen over VHDL wordt de syntax en de grammatica van de hardware beschrijvingstaal VHDL geleerd. Verder worden de basisprincipes, van analyse en ontwerp van combinatorische logische schakelingen met behulp van een Integrated Development Environment (IDE) behandeld waarbij gebruik wordt gemaakt van VHDL. Bij de opdrachten wordt gewerkt met VHDL op een FPGA-prototype board.



In de lessen worden de basisprincipes en de grammatica van de taal VHDL aangeleerd en toegepast.

### 1.2 Leerdoelstellingen

De doelstellingen van de lessen zijn beschreven in de leeruitkomsten zoals in het Opleidingsstatuut en het Onderwijs en examenreglement van de voltijd bacheloropleiding Embedded Systems Engineering in de module Embedded Hardware Engineering – tabel 8. (HAN, 2024)

- De student rekent met getallen in verschillende talstelsels, kent verschillende codeersystemen en rekent aan digitale logica.
- De student kent de basis van het beschrijven en modelleren van programmeerbare logica met behulp van de concurrent- en sequentiële syntax van de hardware beschrijvingstaal VHDL.

- De student test met behulp van de beschikbare synthese- en analyse- hulpmiddelen de logica die is gemodelleerd.

Beoordelingscriteria: De student:

- Rekent in verschillende talstelsels door getallen naar deze talstelsels te converteren.
- Kent de eigenschappen van ASCII, BCD en GRAY-code en past deze coderingssystemen toe.
- Herkent booleaanse logica. Door gebruik te maken van de wetten van de Morgan en de booleaanse rekenregels vereenvoudigt de student deze logica (logical sufficiency).
- Kent de verschillen tussen analoge en digitale signalen en kan beoordelen of een digitaal signaal beïnvloed kan worden door externe bronnen zoals ruis.
- Herkent digitale logische basis bouwstenen en kan deze gebruiken in combinatorische en sequentiële schakelingen. Daar waar timing noodzakelijk is kan de student timing diagrammen interpreteren en maken.
- Kent de concurrent syntax van de taal VHDL.
- Zet een schakeling om in VHDL op basis van gefundeerde keuzes aan de hand van user requirements, debugged en test deze schakeling met behulp van timing-diagrammen en licht de werking toe.
- Begrijpt de verschillen tussen Hardware Description Language en een sequentiële programmeertaal zoals C en C++ en weet welke taal hij waar kan gebruiken.
- Kent de sequentiële syntax van de taal VHDL, en kan deze gebruiken bij finite state machines, clockgeneratie en manipulatie, en het parsen van seriële communicatie.
- Genereert testvectoren voor het ontwerp, gebruikmakend van foutmodellen en test de ontworpen logica op basis van een gegeven specificatie met behulp van test-benches.
- Rapporteert zakelijk en efficiënt de onderbouwing en evaluatie van, en zelfreflectie op de ontworpen, geïmplementeerde en geteste logica.

### 1.3 Lesaanpak

De lessen worden georganiseerd volgens de [Flipped Classroom \(FC\)](#) aanpak. In eenvoudige bewoordingen werkt FC als volgt: *Voorafgaand aan de les*, bestudeert de student de lesonderwerpen met behulp van de lesmaterialen, oefeningen en tentamens. Tijdens de les worden problemen opgelost en wordt de theorie besproken aan de hand van vragen van de studenten. Na de les bestudeert de student zijn aantekeningen, oefent door het oplossen van oefen vragen en neemt het wekelijkse tentamen af.

#### 1.3.1 Flipped Classroom

Figuur 1 toont de activiteiten in FC die in deze paragraaf worden beschreven

Mandatory Pre-class activities		Planned In-class activities		Recommended Post-class activities	
Activity		Activity		Activity	
1	Study theory	1	(Additional) theory on topic by teacher	1	Review theory
2	Prepare questions on theory for class			2	Make weekly exam
3	Reproduce examples from the book	2	Discuss questions from students on theory studied		
4	Make exercises for that week	3	Work on exercises		

Figuur 1: Activiteitenoverzicht bij Flipped Classroom

#### 1.3.1.1 Voorbereidingen voor de les

Voorafgaand aan de les voert de student de volgende verplichte activiteiten uit:

1. Bestudeer de theorie: bestudeer de paragrafen uit het boek voor de les van die week. Als hulpmiddel kan een student een uittreksel maken van de paragrafen in voorbereiding op het wekelijkse tentamen.



De student kan 2 video's bekijken waarin de paragrafen van die week worden besproken. Een video duurt tussen de 15 en 30 minuten (totaal 1 uur). De video's kunnen worden gekeken met behulp van de link op Onderwijs Online. De presentatie die is gebruikt bij de video kan ook via Onderwijs Online worden gedownload.

2. Bereid vragen voor: Tijdens het studeren kan de student vragen voorbereiden ter verduidelijking in de les. Het is handig om voorafgaand aan de les de vragen aan de docent te sturen zodat hij de vraag kan meenemen bij de lesvoorbereiding.
3. Oefen met de voorbeeldvragen in het boek: Het leerproces wordt versterkt door actief met de stof bezig te zijn. Met het maken van oefenopgaven krijg je beter inzicht en gevoel bij het onderwerp.
4. Maak de extra oefeningen die voor deze week gepland zijn: Start zo snel mogelijk met de oefen vragen. Bij de oefenvragen krijg je ook de antwoorden zodat je het resultaat van je week kan controleren.



Om het leren te versterken kan je samenwerken met een of meerdere medestudenten. Je oefent extra door aan elkaar uit te leggen hoe je de opgave hebt gemaakt. Samenwerken kan fysiek op school maar ook online met behulp van Teams.

#### 1.3.1.2 Lesactiviteiten

In de les vinden geplande en ongeplande activiteiten plaats:

1. Bediscussieer klassikaal de vragen van studenten over de theorie van die week.
2. (Aanvullende) theorie over het onderwerp van deze week door de docent. Gebaseerd op de voortgang van de klas of op initiatief van de docent wordt er theorie behandeld aanvullend op wat in het boek staat.
3. Wanneer er tijd is: Extra tijd in de les wordt gebruikt om te werken aan oefen opgaven.

#### 1.3.1.3 Activiteiten na de les

Na de les bestudeert een student de theorie die is besproken in de les. Op deze wijze wordt het leerproces versterkt.



Een belangrijk onderdeel van de lessen is dat na de les het wekelijkse examen wordt gemaakt.

### 1.4 Digitaal lesgeven

Alhoewel de lessen worden aangeboden op de HAN is er de mogelijkheid dat door invloeden van buitenaf deelnemen aan fysieke lessen niet mogelijk is. De Coronapandemie of stakingen in het openbaar vervoer zijn voorbeelden waarbij studenten niet fysiek kunnen deelnemen in het klaslokaal. De lessen kunnen worden aangepast om het programma doorgang te kunnen laten vinden. Bij online lessen of voor het volgen van de lessen via het internet wordt gebruik gemaakt van Microsoft Teams. Voor een prettige beleving van online lessen moeten de aanwijzingen in de volgende paragrafen opgevolgd worden.

#### 1.4.1 Online lessen

Alle lessen worden aangeboden in “hybride vorm”. Dit betekent dat elke les is te volgen op Teams zodat elke student live kan deelnemen aan de les.

Gedurende de les moeten studenten die online deelnemen **hun webcam gedurende de hele les ingeschakeld houden**. Door de webcam aan te hebben creëren we een open sfeer waarin het uitwisselen van kennis en ervaring de standaard kan zijn.

#### 1.4.2 Opnames

Alle lessen worden opgenomen op Teams zodat ze teruggekeken kunnen worden door de studenten. Doordat een student deelneemt aan een online les stemt hij er mee in dat de les wordt opgenomen. Wanneer een student er niet mee instemt dat de les wordt opgenomen, dan neemt deze niet deel aan de online lessen. De student kan dan de opgenomen les achteraf bekijken.

### 1.5 Lesmaterialen

De volgende materialen worden gebruikt bij de lessen:

1. PowerPointpresentaties (Zie het studieschema)
2. Oefen opgaven.

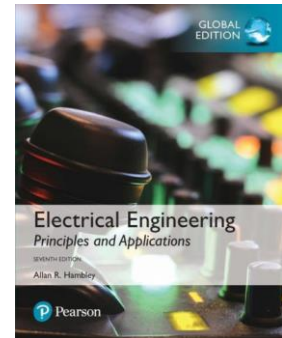


### 1.5.1 Documentatie

De lessen maken gebruik van het volgende boek dat de student in bezit moeten hebben:

#### **Voor sprint 1: (verplicht)**

*Electrical Engineering* door Allen Hambley, 7<sup>e</sup> Editie,  
ISBN-13: 978-0-273-79325-0

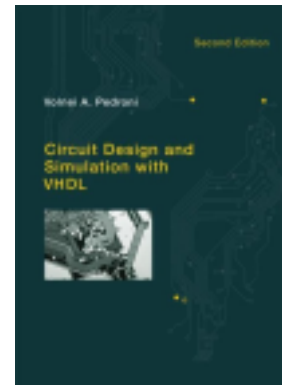


**Een van de twee volgende boeken is verplicht.**

#### **Voor Sprint 2 en 3: (Voorkeur)**

Circuit Design and Simulation Using VHDL, 2nd Edition, Copyright 2010 MIT Press, by Volnei A. Pedroni;  
ISBN 978-0-262-01433-5.

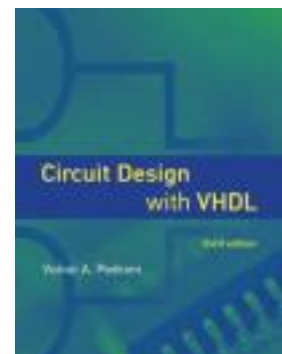
Beschikbaar via deze link: <https://www.bol.com/nl/nl/p/circuit-design-and-simulation-with-vhdl/1001004009894657/>  
(gecontroleerd op 15-1-2024).



#### **Voor Sprint 2 en 3: (Ook toegestaan)**

Circuit Design with VHDL, 3rd edition, by Volnei A. Pedroni;  
ISBN/EAN 9780262042642

Beschikbaar via deze link: <https://www.bol.com/nl/nl/p/circuit-design-with-vhdl/9200000107927138/>  
(gecontroleerd op 15-1-2024).



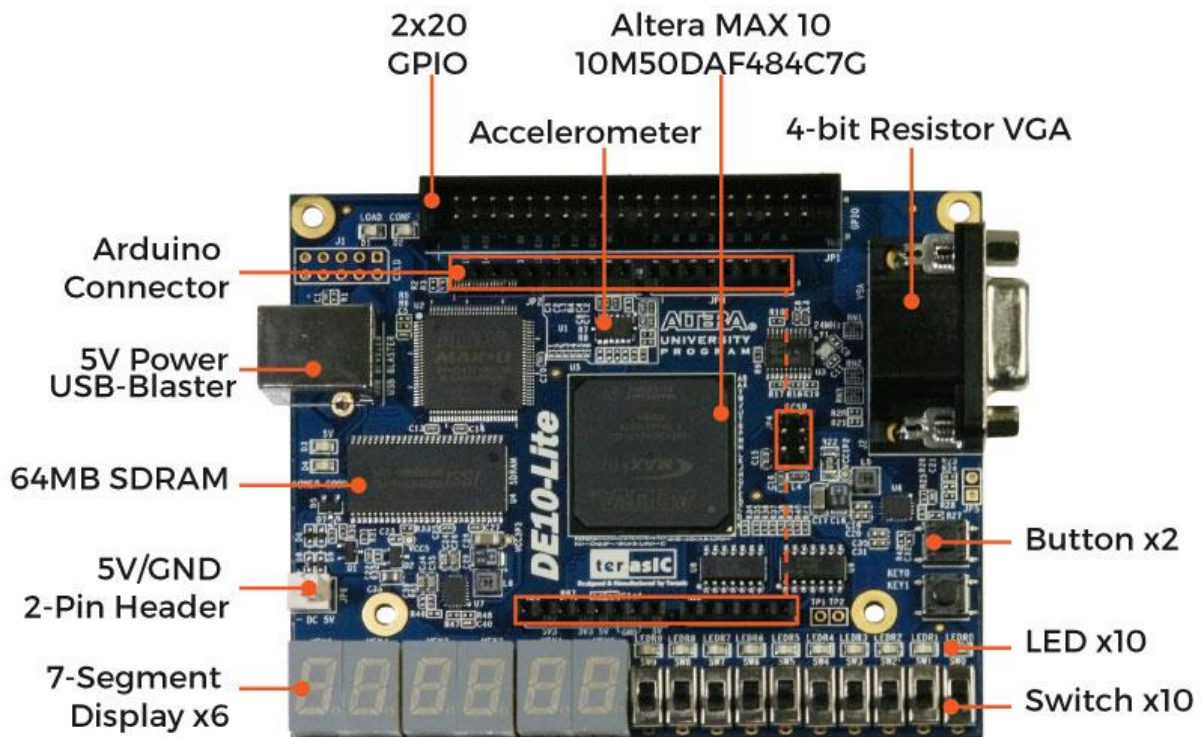
Wanneer dit boek wordt gebruikt moet gebruik worden gemaakt van de vertaaltabel waarmee de hoofdstukken worden vertaald naar de hoofdstukindeling van editie 2.

### 1.5.2 Hardware

Bij de SOC-lessen wordt specifieke hardware gebruikt. In de volgende hoofdstukken wordt benoemd welke hardware dit is en hoe deze verkregen kan worden.

#### *1.5.2.1 Terasic DE10-lite FPGA-prototype board.*

De DE10-Lite is een FPGA-prototype board dat gebruik maakt van de Altera (Intel) MAX10 FPGA. Naast de 7-segments displays, schakelaars, drukknoppen en LEDs beschikt dit board over een Arduino-form factor connector en een accelerometer.



Figuur 2: DE10-Lite FPGA-prototype board

#### 1.5.2.1.1 Aanschaffen via HAN ARLE

Het DE10-Lite board kan worden aangeschaft via HAN ARLE (via Jelle Ellemans) voor € 120,-.

#### 1.5.2.1.2 Aanschaffen via TerasIC

Het DE10-Lite board kan worden aangeschaft bij TerasIC voor \$ 82,- (academic price). Volg de volgende [link](#) houdt een creditkaart bij de hand en houd rekening met +3 weken voordat het board geleverd wordt.

#### 1.5.2.1.3 Aanschaffen van een tweedehands bord van studenten

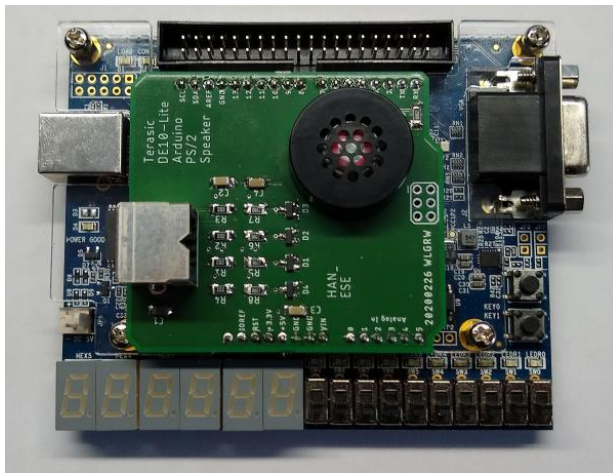
Sommige studenten willen hun DE10-Lite board verkopen. Daarvoor kan je contact zoeken met 3<sup>e</sup> of 4<sup>e</sup> jaar studenten voltijd of met deeltijd studenten die de module D-H-EHE al hebben gevolgd.

Wil je hier gebruik van maken dan kan de docent je in contact brengen met studenten die hun DE10-Lite board willen verkopen.

Registreer jezelf met behulp van het volgende [formulier](#). Studenten die zich aanmelden worden door de docent in contact gebracht met Studenten die hun board willen verkopen. De docent is verder niet betrokken of aanspreekbaar bij de verkoop.

#### 1.5.2.2 Arduino PS/2 Speaker shield voor DE10-Lite

Om een PS/2 compatible keyboard aan te kunnen sluiten op het DE10-Lite board, is een shield een ontwikkeld met een Arduino form factor. Dit shield biedt een PS/2 compatible mini-din female connector en een speaker. Het shield biedt alle noodzakelijke interfacing van en naar het DE10-Lite board.



Figuur 3: PS/2 shield op De10-Lite FPGA board

De documentatie van het PS/2 shield is onderdeel van de git repository die gebruikt wordt bij opdracht 2: <https://gitlab.com/wlgrw/han-soc-assignment-vhdl-piano/-/tree/master/hardware>

#### 1.5.2.2.1 Verkrijgen PS/2 shield

The PS2-keyboard shield wordt aangeboden als soldeerkit die de student zelf kan solderen. Afhankelijk van de wens van de klas, kan een workshop georganiseerd worden waarin de kit gesoldeerd kan worden op de HAN.

De prijs van de kit is 14 euro. Deze is niet gesoldeerd en bestaat uit alleen de componenten en de PCB. Wanneer de student niet over de juiste middelen beschikt om zelf te solderen kan de kit voor 5 euro gesoldeerd worden.

Voor het bestellen zal een link beschikbaar gesteld worden naar een bestelformulier.

#### 1.5.3 Software

Tijdens de lessen wordt gebruik gemaakt van verschillende programma's. Deze worden tijdens de les benoemd en toegelicht en zijn terug te vinden in de les planning.

Software die gebruikt wordt is:

- Intel® Quartus® Prime Lite versie 19.1 (verplicht)
- Git via GitLab
- Git voor Windows
- TortoiseGit

#### 1.6 Les planning

De H-EHE-SOC lessen bestaan uit theorielessen en opdrachten.

De lesplanning van de theorielessen is beschreven in het document "2223\_H-EHE-SOC\_LesPlanning\_xxx.pdf" dat kan worden gedownload vanaf Onderwijs Online.



De namen van de verschillende documenten voor deze lessen worden voorafgegaan door het *academische jaar* en gevolgd door een *versienummer* als onderdeel van de bestandsnaam. De tweede versie van de lesplanning in het academische jaar 2022-2023 krijgt zo de naam: "2232\_H-EHE-SOC\_LesPlanning\_002.pdf"

### 1.6.1 Theorielessen

Tijdens de theorielessen wordt de theorie onderwezen. Tegelijkertijd kunnen in de theorielessen onderwerpen behandeld worden die voor de opdrachten van belang zijn. Deze onderwerpen kunnen door zowel de student als de docent naar behoefte worden voorgesteld.

De SOC-lessen zijn opgedeeld in 3 sprints waarin de volgende opbouw wordt gerealiseerd.

#### 1.6.1.1 *Sprint 1: Logic Circuits*

In sprint 1 wordt de basis van digitale techniek en logische schakelingen behandeld ter voorbereiding op het ontwerp van digitale schakelingen in de hardware beschrijvingstaal VHDL.

Uitgangspunt is dat de student basiskennis heeft van deze onderwerpen. Hierdoor hebben de lessen het karakter van herhaling en het opfrissen van kennis.

#### 1.6.1.2 *Sprint 2: Concurrent Logic in VHDL*

In Sprint 2 wordt de grammatica van de hardware beschrijvingstaal VHDL geleerd. Hierbij staat het testen van ontworpen logica centraal waarbij gebruik wordt gemaakt van de mogelijkheden die de taal VHDL biedt.

Bij het ontwerpen en implementeren van logica wordt alleen gebruik gemaakt van de concurrent elementen in VHDL.

In de opdracht komt de kennis van sprint 1 samen met de kennis van sprint 2.

#### 1.6.1.3 *Sprint 3: Sequential Logic in VHDL*

In deze sprint wordt de kennis van VHDL die is opgedaan in sprint 2 uitgebreid met het sequentiële deel van VHDL.

In de opdracht wordt gewerkt met statemachines en frequentiedelers.

### 1.6.2 Opdrachten

De SOC-lessen maken gebruik van 2 opdrachten die elk afzonderlijk worden beoordeeld. Studenten werken in vaste koppels samen aan deze opdrachten. Deze vaste koppels leveren ook gezamenlijk de beroepsproducten op.

De opdrachten zijn:

1. Het met behulp van VHDL realiseren en testen van een rekenkundige unit (ALU) op het FPGA-prototype board.
2. Het met behulp van VHDL realiseren en testen van een piano op het FPGA-prototype board.

#### 1.6.2.1 *VHDL-code en Quartus-projecten*

De VHDL-code en Quartus projecten zijn een verzameling van opdrachten per projecten, die beschreven zijn in de documenten: "SOC\_Assignment-1\_xxx.docx" en "SOC\_Assignment-2\_xxx.docx" die gedownload kunnen worden vanaf Onderwijs Online.

De opdrachten worden door de studenten in koppels uitgevoerd. De werkzaamheden worden buiten de SOC-lessen uitgevoerd en ondersteund vanuit de theorielessen.

Ten behoeve van het versiebeheer wordt verplicht gebruik gemaakt van Git en GitLab.

De VHDL-code die wordt gerealiseerd in de opdrachten, maken gebruik van een verplichte template die beschikbaar wordt gesteld voor deze opdrachten. De template is afhankelijk van het type hardware dat in de SOC-lessen wordt gebruikt. De template is een public git repository in GitLab die

de student moet klonen in een eigen private git-repository op het platform van GitLab (<https://gitlab.com/>).



Bij het aanmaken van de git repository moet gebruik worden gemaakt van de volgende naamconventie:

“SOC\_<studentNaam1>\_<studentNummer1>\_<studentNaam2>\_<studentNummer2>”

Hierbij is:

- <studentNaam1> de naam van de eerste student in lowerCamelCase<sup>1</sup>
- <studentNummer1> het studentnummer van de eerste student
- <studentNaam2> de naam van de tweede student in lowerCamelCase
- <studentNummer2> het studentnummer van de tweede student

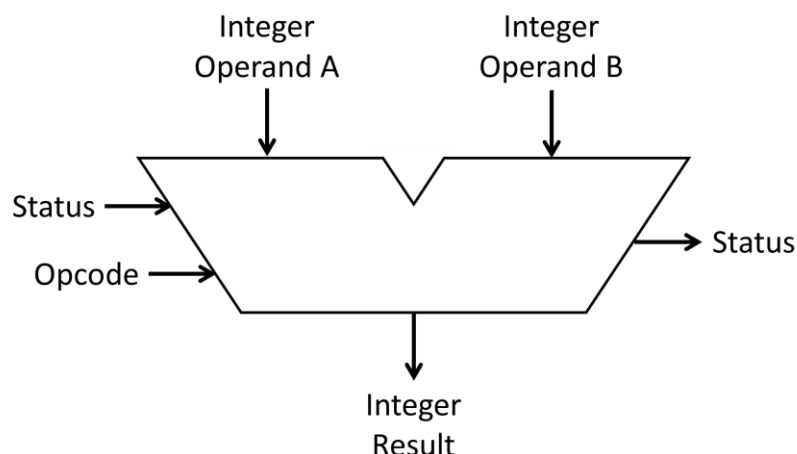
Eventueel kan op dezelfde wijze een derde student worden toegevoegd.

- Het maken van een nieuw project in GitLab met gebruikmaking van een bestaande git repository is beschreven in [https://docs.gitlab.com/ee/user/project/import/repo\\_by\\_url.html](https://docs.gitlab.com/ee/user/project/import/repo_by_url.html)
- De repository van de studenten moet **private** gemaakt worden volgens deze documentatie: [https://docs.gitlab.com/ee/user/admin\\_area/settings/visibility\\_and\\_access\\_controls.html#configure-project-visibility-defaults](https://docs.gitlab.com/ee/user/admin_area/settings/visibility_and_access_controls.html#configure-project-visibility-defaults)
- De beoordelaar van de VHDL-code moet de rechten van **reporter** worden toegewezen. De gebruikersnaam van de beoordelaar wordt in de lessen bekendgemaakt aan de studenten

#### 1.6.2.2 Opdracht 1: Arithmetic Logic Unit in VHDL

Een arithmetic logic unit (ALU) is a combinatorische digitale schakeling die rekenkundige en bitwise-logische operatoren kan uitvoeren op digitale getallen. Een ALU is een fundamentele bouwsteen voor verschillende soorten computers. Een voorbeeld is de centrale processing eenheid (CPU) van een computer (Wikimedia, 2022).

Een ALU is het onderdeel van de microprocessor die optelt, aftrekt, vermenigvuldigt, deelt en nog veel meer. De ALU is het onderdeel dat deze operaties werkelijk uitvoert.



Figuur 4: Functionele representatie van een ALU

<sup>1</sup> [https://en.wikipedia.org/wiki/Camel\\_case](https://en.wikipedia.org/wiki/Camel_case)

De inputs van een ALU is de data die bewerkt wordt en worden *operands* genoemd. De code die de operatie representeert die moet worden uitgevoerd heet de Operation CODE of *opcode*.

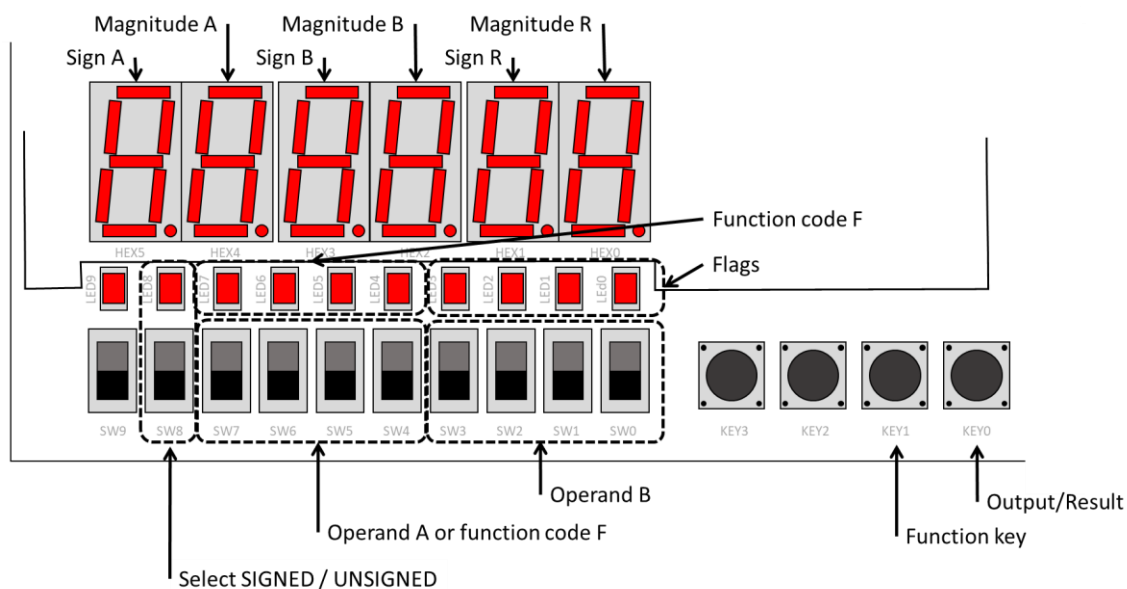
Op de output van de ALU's staat het *resultaat* van de uitgevoerde operatie. De ALU heeft ook *status inputs* die informatie draagt over de voorgaande operatie en *status outputs* die het resultaat vormen van de huidige operatie.

Een basis ALU heeft drie parallelle databussen waarvan er 2 de *operands* (A en B) zijn en een het resultaat (R). Elke databus draagt de informatie van een binair geheel getal. Normaal gesproken zijn de databussen allemaal even groot en komen ze overeen met de woordgrootte van het interne circuit (de CPU).

De *opcode* is een parallelle bus waarop de selectie code van de operatie wordt ingegeven. De *opcode* is een enumerated getal dat verwijst naar de rekenkundige of logische bewerking die door de ALU moet worden uitgevoerd.

De status output bestaat uit verschillende individuele signalen die aanvullende informatie geven over het resultaat van de huidige ALU-operatie.

De ALU maakt gebruik van de in- en output van het FPGA ontwikkelbord. In Figuur 5 wordt de bediening getoond voor een DE10-Lite board. Wanneer gebruik wordt gemaakt van de DE0-CV wordt de bediening beschreven in een specifiek voor dat bord gepubliceerde handleiding.



Figuur 5: User interface ALU op de De10-Lite

Bij deze opdracht wordt een 4-bit ALU gerealiseerd en beperken we ons tot een gereduceerde instructieset.

De opdracht wordt:

- Uitgevoerd met behulp van de IDE Quartus van Intel. In Quartus wordt de ALU gecodeerd en getest met behulp van waveforms.
- Aangeboden in de vorm van een template die als Git repository kan worden gedownload.



Voor deze opdracht wordt gebruik gemaakt van de volgende repository:

- DE0-10-Lite: <https://gitlab.com/wlgrw/han-soc-assignment-vhdl-alu>



### 1.6.2.3 Opdracht 2: Piano in VHDL

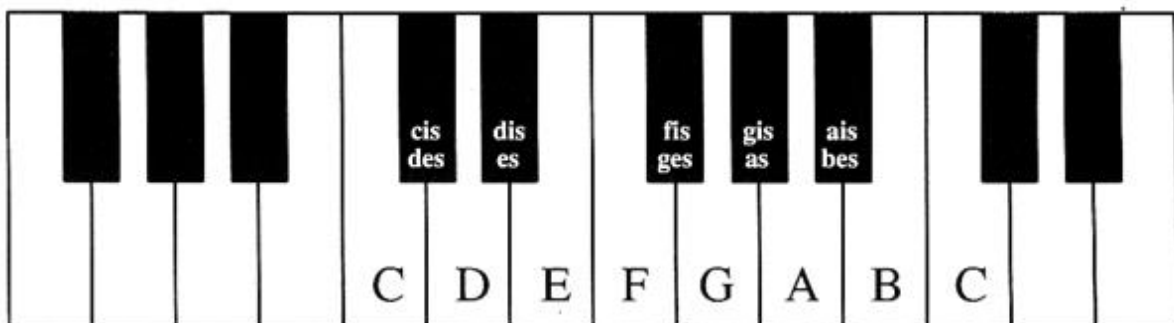
In deze opdracht gaan we een elektronische piano maken met behulp van een PS/2-toetsenbord, een TerasIC DE0-CV FPGA-prototypebord en een kleine luidspreker.

Op de DE0-CV moet een PS/2-toetsenbord worden aangesloten. Een kleine speaker, die op de GPIO-poorten wordt aangesloten, zal de toetsaanslagen op de piano hoorbaar maken.

Het PS/2-toetsenbord vervangt een normaal instrumenttoetsenbord. Aangezien het PS/2-toetsenbord dat we voor deze opdracht gaan gebruiken geen witte en zwarte toetsen heeft en de toetsen niet gerangschikt zijn zoals op een instrument zoals een piano of orgel, zullen we de toetsen van het PS/2-toetsenbord zo gebruiken dat het overeen komt met een klavier van een piano. Bij deze indeling staan de zwarte toetsen op de rij met de cijfertoetsen en de witte toetsen op de rij met de letters "QWERTY".

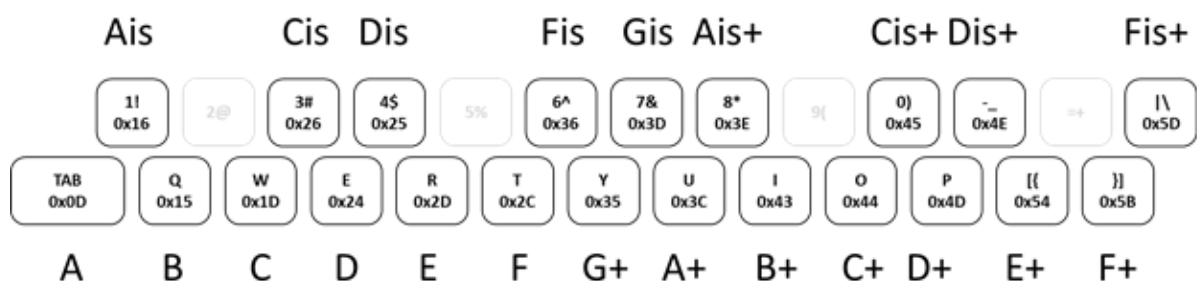
Bij een echt instrument kan de muzikant meerdere tonen tegelijk indrukken maar dat zal met deze uitvoering van de piano niet mogelijk zijn. Daarom beoordelen we de student niet op zijn muzikale kwaliteiten.

Het PS/2-toetsenbord vervangt een normaal klavier, zoals weergegeven in afbeelding 3. In deze afbeelding zijn de primaire noten geschreven op de zwarte en witte toetsen.



Figuur 6: Pianoklavier met noten

Het PS/2-toetsenbord dat we voor deze opdracht gaan gebruiken heeft geen witte en zwarte toetsen en de toetsen zijn niet gerangschikt zoals op een instrument zoals een piano of orgel. Daarom zullen we de toetsen van het PS/2-toetsenbord gebruiken op een manier die het instrumenttoetsenbord weergeeft zoals gepresenteerd in Figuur 7.

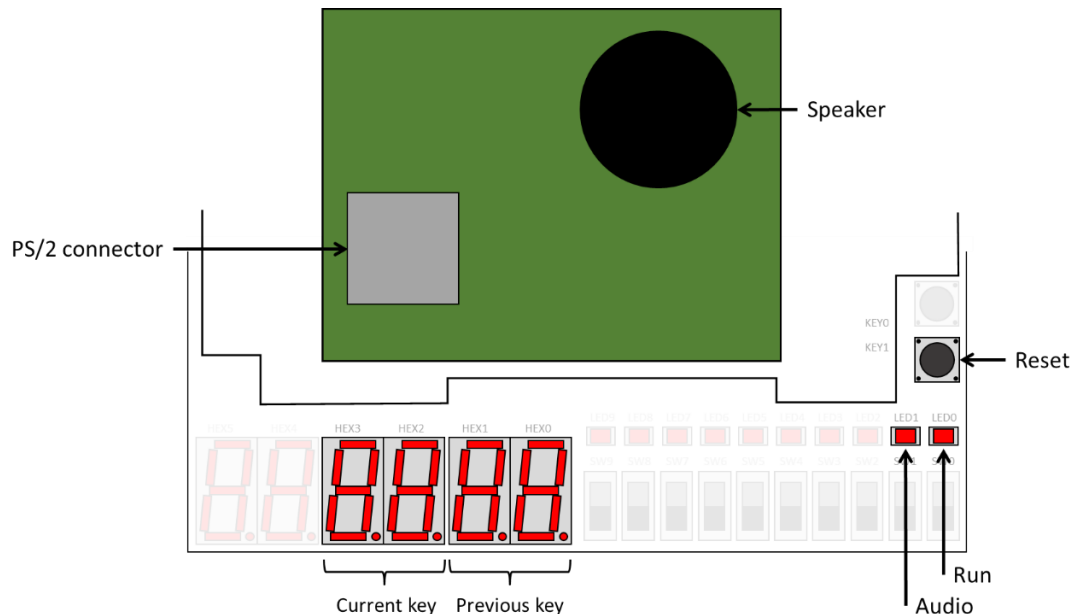


Figuur 7: Noten en toetsen plaatsing op het PS/2 keyboard.

Bij deze indeling staan de zwarte toetsen op de rij met de cijfertoetsen en de witte toetsen op de rij met de letters "QWERTY".

Elke toetsaanslag, wanneer toegewezen aan de toetsen die een toon genereren, zal resulteren in de overeenkomstige toon die op de luidspreker te horen is. Audio wordt ook weergegeven op LED1. De hexadecimale waarde van de scancode van de toets die wordt ingedrukt, wordt weergegeven op HEX2 en HEX3, terwijl de hexadecimale waarde van de scancode die de vorige toets vertegenwoordigt, wordt weergegeven op HEX0 en HEX1.

KEY1 van het DE0-CV-bord reset de VHDL-piano. Een reset wordt aangegeven wanneer LED0 uit is.



Figuur 8: User interface van de VHDL piano op het DE0-CV FPGA board

De opdracht wordt:

- Uitgevoerd met behulp van de IDE Quartus van Intel. In Quartus wordt de VHDL-piano gecodeerd en getest met behulp van testbenches.
- Aangeboden in de vorm van een template die als Git repository kan worden gedownload.



Voor deze opdracht wordt gebruik gemaakt van de volgende repository:

- DE0-10-Lite: <https://gitlab.com/wlgrw/han-soc-assignment-vhdl-piano>

## 1.7 Tentamen en beoordeling

### 1.7.1 Sprint 1

Aan het einde van het deel "Logic circuits" kunnen studenten een online tentamen afnemen om hun kennisniveau te testen. Deze toets wordt niet beoordeeld en is informatief (formatief).



Dit online tentamen is een "open-boek" tentamen. Dit betekent dat er geen proctoring of surveillance plaats vindt. Met dit tentamen toon je aan wat jouw kennis is van de theorie.

Het online wordt gemaakt in ANS (ANS, 2022), het digitale examensysteem dat gebruikt wordt door de HAN. De studenten worden door de docent toegevoegd aan het systeem.

Voor deze online tests is de student aangemeld bij ANS. De docent heeft deze registratie uitgevoerd waarna de student een uitnodiging ontvangt om de registratie af te ronden. Na deze registratie kan de student deelnemen aan het online tentamen die voor hem klaargezet worden





Zorg ervoor dat je een werkend account bij ANS hebt. Wanneer dat niet het geval is neem dan contact op met de docent.

ANS is te bereiken via deze link: [https://ans.app/users/sign\\_in](https://ans.app/users/sign_in)

### 1.7.2 Sprint 2

Aan het einde van sprint 1 "Concurrent Logic in VHDL", leveren studenten de programmacode van het project met hun ALU in VHDL in. De VHDL-code in dit project wordt beoordeeld met een cijfer.

#### 1.7.2.1 Beoordeling

De beoordeling van de eerste opdracht is gebaseerd op de ingediende project-code.

Het cijfer van de eerste opdracht is samengesteld uit:

- 50% - Code kwaliteit van de door de student ingeleverde VHDL-code.
- 50% - Kwaliteit van de implementatie van de functies in de door de student ingeleverde VHDL-code.

De beoordeling vindt plaats aan de hand van beoordelingscriteria die zijn vastgelegd in rubrieken. Deze rubrieken kunnen worden gedownload van Onderwijs Online.

Het cijfer voor de tweede periode (sprint 3) wordt berekend met de volgende formule:

$$1 + 90\% \cdot (50\% \cdot \text{codeQualityAssignment1} + 50\% \cdot \text{implementationQualityAssignment1})$$

De registratie van de beoordeling wordt uitgevoerd door de beoordelaar waarbij gebruik wordt gemaakt van een beoordelingsformulier. Het beoordelingsformulier kan gedownload worden van Onderwijs Online.

### 1.7.3 Sprint 3

Aan het einde van sprint 2 "Sequential Logic in VHDL" moeten studenten:

- De programmacode van hun Piano in VHDL indienen. The VHDL-code van dit project wordt beoordeeld met een cijfer.
- Deelnemen aan een mondeling tentamen waarin het project en de individuele kennis van de student wordt beoordeeld.

#### 1.7.3.1 Mondeling tentamen

Studenten worden in groepen beoordeeld in een mondeling tentamen aan het einde van sprint 3 van de SOC-lessen. Een mondeling tentamen neemt ca. 20 minuten waarin:

- De student antwoorden geeft op vragen naar aanleiding van de VHD-code van het 2<sup>e</sup> project dat is ingediend waarmee de student zijn kennis van VHDL aantoont.
- De student demonstreert de werking van het ingeleverde project tijdens het mondelinge tentamen.

De mondelinge tentamens worden gehouden in de 2 tentamenweek aan het einde van periode 2 waarin de SOC-lessen worden gegeven.



De datum waarop de mondelinge tentamens worden gehouden, worden aangekondigd door de docent.

De studenten kunnen zelfstandig een reservering doen voor een van de beschikbare tijdstippen met behulp van het reserveringssysteem Bookings. De volgende werkwijze wordt gevolgd:

- Per groepje maakt één student de reservering voor zijn groepje. Deze student is er verantwoordelijk voor dat de reservering gegevens met zijn mede student wordt gedeeld.
- Bij de reservering moet de student gebruik maken van zijn HAN student emailadres.
- Tijdens de reservering moeten de voor- en achternamen alsook het studentnummer van alle studenten die in de groep zitten worden doorgegeven.
- De reserveringen zijn “First come, first serve!”



Alleen studenten die voor de deadline hun beroepsproducten hebben ingeleverd via HandIn worden geaccepteerd voor een mondeling tentamen.

#### 1.7.3.2 Beoordeling

De beoordeling van de tweede opdracht is gebaseerd op de ingediende project-code en de prestatie van de individuele student tijdens het mondeling.

Het cijfer van de tweede opdracht is samengesteld uit:

- 40% - Code kwaliteit van de door de student ingeleverde VHDL-code.
- 60% - Kwaliteit van de antwoorden die door de student zijn gegeven tijdens het mondelinge tentamen.

De beoordeling vindt plaats aan de hand van beoordelingscriteria die zijn vastgelegd in rubrieken. Deze rubrieken kunnen worden gedownload van Onderwijs Online.

Het cijfer voor de tweede periode (sprint 3) wordt berekend met de volgende formule:

$$1 + 90\% \cdot (40\% \cdot \text{codeQualityAssignment2} + 60\% \cdot \text{oralGrade})$$

De registratie van de beoordeling wordt uitgevoerd door de beoordelaar waarbij gebruik wordt gemaakt van een beoordelingsformulier. Het beoordelingsformulier kan gedownload worden van Onderwijs Online.

#### 1.7.4 Indienen van opdrachten

Beide projecten moeten worden onderhouden in het versie beheerssysteem Git op het GitLab platform.

Beroepsproducten moeten worden ingeleverd in HandIn: <https://handin.han.nl>



Zorg ervoor dat je een groep aangemaakt in handIn voor jouw groepje waarvoor de beroepsproducten worden ingeleverd

- In handin, zoek naar “t-DLT-HM-EHE-SOC”.
- Upload het ZIP-bestand met de git repository van je VHDL-project
- Geef de URL van de Git repository op waarvan je de release hebt ingeleverd namens jullie groep.
- Upload het product document
- Vul eventuele opmerkingen in
- Dien je producten in ter beoordeling.

#### 1.7.5 Deadlines



Projecten voor beide opdrachten dienen ingediend te worden **vóór de deadline** die in de planning is vastgelegd. Een vertraging bij het indienen zal resulteren in een vertraging van de beoordeling en beoordeling.

#### 1.7.6 Herkansingen

Herkansingen van een beoordeling of het mondelinge tentamen moeten worden gepland in overleg met de beoordelaar.

Wanneer een student een herkansing nodig heeft zal op het initiatief van de student, in nauw overleg met de beoordelaar opnieuw afgesproken worden. Dit is afhankelijk van de situatie van de student en welke voortgang de student heeft gemaakt.

## Bibliography

ANS. (2022, 06 17). *The satisfying grading platform*. Opgehaald van ANS: <https://ans.app/landing>

HAN. (2024, 1 15). *BIJLAGE bij het Opleidingsstatuut en Onderwijs- en Examenregeling van de deeltijdse bacheloropleidingen van HAN University of Applied Sciences: Embedded Systems Engineering*. Opgehaald van OER: <https://oer.han.nl/28347335-625a-4298-ae19-8a4febce815d>

Wikimedia. (2022, 02 18). *Arithmetic logic unit*. Opgehaald van Wikipedia: [https://en.wikipedia.org/wiki/Arithmetic\\_logic\\_unit](https://en.wikipedia.org/wiki/Arithmetic_logic_unit)

## Appendix

### Vertaaltabel Pedroni 2<sup>nd</sup> edition naar 3<sup>rd</sup> edition

Voor de studenten die Digital Circuit Design with VHDL, 3e editie hebben gekocht, geeft deze tabel aan hoe de alinea's zijn toegewezen aan Circuit Design and Simulation Using VHDL, 2e editie.

Houd er rekening mee dat dezelfde theorie in de 3e editie anders kan worden gepresenteerd dan in de 2e editie.

*Tabel 1: Vertaaltabel tweede naar derde editie*

2nd Edition			3rd edition	
Par.	Title		Par.	
1	Introduction	-	5	
2	Code Structure	-	6	
3	Data Types	-	7 (6)	
4	Operators and Attributes	-	9	
5	Concurrent Code	-	10	
6	Sequential Code	-	12	