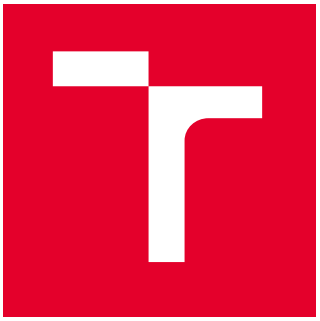


BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF ELECTRICAL AND ELECTRONIC TECHNOLOGY

ÚSTAV ELEKTROTECHNOLOGIE

ANALYSIS OF THE OPTIMIZATION OF THE DIGITAL PART OF THE HW USB CHARGER

ANALÝZA OPTIMALIZACE DIGITÁLNÍ ČÁSTI HW USB NABÍJEČKY

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR — **Matej Dinis**
AUTOR PRÁCE

SUPERVISOR — **doc. Ing. Petr Vyroubal, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2025

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

Bachelor's Thesis

Bachelor's study program **Microelectronics and Technology**

Department of Electrical and Electronic Technology

Student: Matej Dinis ***ID:*** 246992

Year of study: 3 ***Academic year:*** 2024/25

TITLE OF THESIS:

Analysis of the optimization of the digital part of the HW USB charger

INSTRUCTION:

Familiarize yourself with the charging technology of Li-Pol and EDLC (Super Cap) and study the existing specific implementation of the digital part of the charging circuit.

Analyze and document the proposed modification in the control of the existing solution, transitioning from a partially software-controlled approach to a fully hardware-autonomous solution. The proposed solution must newly support the JEITA charging standard.

Then, design and specify the system architecture of the digital part of the circuit for fully autonomous hardware-based charging control. The solution must also include a simple register map allowing for trimming and parameterization of the charging process from internal memory, as well as status registers enabling feedback evaluation of the charging process, control, and monitoring.

Implement the digital control and monitoring, part of the charging process at the RTL level and demonstrate its correct functionality.

RECOMMENDED LITERATURE:

Dle pokynů vedoucího bakalářské práce.

JEITA - https://home.jeita.or.jp/upload_file/20210407112651_JbDgUSWGiB.pdf

Návrh čítačů - https://www.dps-az.cz/clanky/id:6116/implementace-citacu-v-cislicovych-systemech

Návrh FSM - Microsoft Word - CummingsSNUG1998SJ_FSM_rev1_1.doc (sunburst-design.com)

Date of project specification: 10.2.2025 ***Deadline for submission:*** 5.6.2025

Supervisor: doc. Ing. Petr Vyroubal, Ph.D.

doc. Ing. Pavel Šteffan, Ph.D.
Chair of study program board

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Mikroelektronika a technologie**

Ústav elektrotechnologie

Student: Matej Dinis ***ID:*** 246992

Ročník: 3 ***Akademický rok:*** 2024/25

NÁZEV TÉMATU:

Analýza optimalizace digitální části HW USB nabíječky

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s technologií nabíjení Li-pol, EDLC (Super Cap) a prostudujte stávající konkrétní implementaci digitální části obvodu pro nabíjení.
Analyzujte a zdokumentujte návrh změny v řízení stávajícího řešení z částečně SW-řízeného na plně HW autonomní řešení. Navrhované řešení musí nově podporovat také JEITA nabíjecí standard.
Poté navrhněte a specifikujte systémový návrh digitální části obvodu pro plně autonomní HW řízení nabíjení. Řešení musí obsahovat také jednoduchou registrovou mapu umožňující trimování a parametrizaci procesu nabíjení z interní paměti a funkci statusových registrů umožňujících zpětné vyhodnocení procesu nabíjení, řízení a kontrolu nabíjecího procesu.
Implementujte digitální část řízení a kontroly nabíjecího procesu na RTL úrovni a prokažte jeho správnou funkci.

DOPORUČENÁ LITERATURA:

Dle pokynů vedoucího bakalářské práce.
JEITA - https://home.jeita.or.jp/upload_file/20210407112651_JbDgUSWGiB.pdf
Návrh čítačů - https://www.dps-az.cz/clanky/id:6116/implementace-citacu-v-cislicovych-systemech
Návrh FSM - Microsoft Word - CummingsSNUG1998SJ_FSM_rev1_1.doc (sunburst-design.com)

Termín zadání: 10.2.2025 ***Termín odevzdání:*** 5.6.2025

Vedoucí práce: doc. Ing. Petr Vyroubal, Ph.D.

doc. Ing. Pavel Šteffan, Ph.D.
předseda rady studijního programu

ABSTRACT

This thesis deals with design of hardware battery charger based on original software solution. The charger is designed as an independent unit that can be used in an application with or without a CPU. Charger supports temperature measurement and four temperature zones with two current values for each phase to be JEITA guideline compliant. The theory describes the construction and charging process of Li-pol and EDLC, and the basic HDL building blocks needed for this design and functional and formal industry standard verification methods. The practice provides example of analog part that can be used, and most importantly, register map definition, block level architecture and state diagrams for automats. Scenarios that verify major charger functions are listed. Also approximate area estimation is presented based on the synthesis result.

KEYWORDS

charging, counter, EDLC, FSM, HDL, JEITA, LINT, Li-pol, register map, verification

ABSTRAKT

Táto práca sa zaoberá návrhom hardvérovej nabíjačky batérií z pôvodného softvérového riešenia. Nabíjačka je navrhnutá ako samostatná jednotka, ktorú možno použiť v aplikácii s procesorom alebo bez neho. Nabíjačka podporuje meranie teploty a štyri teplotné zóny s dvoma hodnotami prúdu pre každú fázu, aby bola v súlade so smernicou JEITA. Teória opisuje konštrukciu a proces nabíjania Li-pol a EDLC a základné stavebné bloky HDL potrebné pre tento návrh a metódy funkčnej a formálnej verifikácie podľa priemyselných noriem. V praxi sa uvádza príklad analógovej časti, ktorú možno použiť a predovšetkým definícia mapy registrov, architektúra na úrovni blokov a stavové diagramy pre automaty. Uvedené sú scenáre, ktoré overujú hlavné funkcie nabíjača. Takisto je uvedený približný odhad plochy na základe výsledku syntézy.

KĽÚČOVÉ SLOVÁ

EDLC, FSM, HDL, JEITA, LINT, Li-pol, mapa registrov, nabíjanie, počítadlá, verifikácia

ROZŠÍRENÝ ABSTRAKT

Každodenné prenosné zariadenia, ktoré sú bežne používané miliónmi ľudí, vyžadujú zdroje elektrickej energie. Najviac používané nabíjateľné batérie sú Li-ion a Li-pol alebo superkondenzátory, hlavne vďaka skvelému pomeru kapacity a váhy. Obrovská nevýhoda oproti starším typom batérií je nabíjanie. Pre bezpečne nabíjanie je potreba predpísané hodnoty napätia, prúdu, teploty a času. Je nevyhnutné dodržiavať odporúčania uvedené výrobcom a JEITA odporučenie.

Cieľom tejto práce je popísať proces nabíjania batérií a EDLC superkondenzátorov. Z pôvodnej softvérovej nabíjačky vytvoriť plne hardvérovú nabíjačku, ktorá má možnosť parametrizovať nabíjací proces v troch fázach, a podporovať JEITA odporučenie o bezpečnom nabíjaní batérií pre rôzne typy batérií.

Teoretický úvod práce je zameraný na konštrukciu a hlavne na proces nabíjania batérií. Je v ňom uvedená základná štruktúra li-ion a li-pol batérií, používané materiály pre katódu, anódu a elektrolyt ich výhody a nevýhody. Hlavnou časťou je proces nabíjania, kde sú popísané základné fázy nabíjania. Pre každú fázu sú uvedené typické hodnoty napätia a prúdu v pomere ku kapacite batérie. A možné problémy čo môžu nastať počas nabíjania. Podobne je popísaná konštrukcia a proces nabíjania pre EDLC, kde tento proces je oproti batériám podstatne jednoduchší. Dôležitou častou procesu bezpečného nabíjania je meranie teploty batérie alebo EDLC, pre to bude využité JEITA odporučenie o bezpečnom nabíjaný. Kde je popísané ako bezpečne nabíjať batérie pri rôznych teplotách, kedy sa ešte môžu batérie nabíjať a kedy už to môže byť nebezpečné.

Posledná časť teórie bola venovaná postupu digitálneho návrhu. Popis jednotlivých základný blokov, ktoré sú využité v tejto práci ako sú. Stavové automaty ich základný abstraktný popis rozdelenie a spôsob návrhu. Taktiež je popísaná registrová mapa pre ukladanie hodnôt, spôsob synchronizácie vstupných signálov a typy čítacou. A hlavne je popísaný spôsob verifikácie digitálnej logiky, funkčne a formálne bežne používané v technickej praxi.

Návrh hardvérovej nabíjačky pozostával hlavne z pochopenia funkcie pôvodnej implementácie, ktorá bola takmer celá softvérová. Analógové vstupy boli ukladané do jedného registru, táto premenná udávala aktuálnu fázu nabíjania. Hodnota bola porovnávaná s predefinovanou hodnotou v programe, podľa toho nastavoval program maximálnu hodnotu napätia. Periodicky taktiež program meral napätie baterky a teplotu. Tieto funkcionality boli rozdelené do samostatných blokov.

Výsledný hardvérový návrh pozostáva z troch častí, každá spĺňa jednu funkciu pôvodnej nabíjačky. Hlavný blok je zodpovedný o celý proces nabíjania synchronizuje vstupné signály z analógovej časti, vyhodnocuje aktuálnu fázu nabíjania, spúšťa časovač a periodické meranie. Hlavný blok obsahuje registrovú mapu, sú v

nej uložené parametre nabíjacieho procesu, maximálne hodnoty časovača pre každú fázu, prúdové limity pre normálne nabíjanie, napäťové limity a štyri teploty a redukovaný prúd, ktorý je potreba pre bezpečné nabíjanie podľa JEITA odporučenia. Taktiež slúži pre uloženie aktuálnych hodnotu vstupov pre vyšší prvok alebo procesor. Ďalší potrebný blok pre správne nabíjanie je časovač, pre normálne nabíjanie ale aj pre JEITA. Posledný blok je stavový automat pre meranie, pomocou SAR algoritmu vyhodnocuje aktuálnu hodnotu napätia baterky, teplotu a USB. Kontroluje merané hodnoty a signalizuje hlavnému stavovému automatu ak je treba redukovať maximálny prúd.

Výsledná hardvérová nabíjačka obsahuje samostatnú registrovú mapu, preto pred integráciou do projektu, bolo potrebné vytvoriť medzi vrstvu s integráciou NVM radiča a modelu pamäte pre načítanie dát do registrovej mapy a jednoduchý stavový automat pre načítanie dát a spustenie nabíjačky. Radič a model boli rovnako ako SAR použité z iného projektu.

V simulácii bol pripojený celý design ku SV-RNM modelu, ktorý modeloval reálnu batériu aj s poruchovými stavmi a analógovú časť nabíjačky. Testovanie scenáre otestovali základné fungovanie nabíjačky aj chybové stavy a taktiež aplikáciu JEITA. Hotový RTL kód bol taktiež skontrolovaný Lint s pravidlami, ktoré sa používajú v priemyselnej praxi.

Ako posledné bola pomocou syntézy zistená približná veľkosť celého design pre možnú integráciu do ďalších projektov. Výsledky syntézy taktiež ukázali neporušenie pravidiel presahu a predstihu klopných obvodov.

DINIS, Matej. *Analysis of the optimization of the digital part of the HW USB charger*. Bachelor's Thesis. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Electrical and Electronic Technology, 2025. Advised by doc. Ing. Petr Vyroubal, PhD

Author's Declaration

Author:	Matej Dinis
Author's ID:	246992
Paper type:	Bachelor's Thesis
Academic year:	2024/25
Topic:	Analysis of the optimization of the digital part of the HW USB charger

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno ....

author's signature*

*The author signs only in the printed version.

ACKNOWLEDGEMENT

I would like to thank the advisors of my thesis, doc. Ing. Petr Vyroubal PhD. and multiple colleagues from ASICentrum Tomáš Tuček, Michal Kajan, Michal Skiba, Jakub Štasný, and others for their valuable comments about the thesis structure and hardware design.

Contents

Introduction **15**

1 Lithium-ion batteries **16**
1.1 Construction and Materials 16
1.2 Charging process 18

2 EDLC **20**
2.1 Construction and Materials 20
2.2 Charging process 21

3 JEITA **23**

4 Digital design hierarchy and principles **25**
4.1 FSM 25
4.2 Register map 27
4.2.1 D-FF based resynchronization 27
4.3 Counters 28
4.4 Verification 29
4.4.1 Simulation 29
4.4.2 Formal 30

5 Tasks for bachelor's thesis **31**

6 Optimization target **32**
6.1 Analogue part 32
6.1.1 Charging 32
6.1.2 JEITA related measurement 32
6.1.3 Digital part interface 33
6.2 Original software 34

7 New block level architecture **35**
7.1 Charger top Level 36
7.1.1 Requirements 36
7.2 Top Level FSM 37
7.2.1 Requirements 39
7.3 Measurement module 40
7.3.1 Requirements 41
7.4 Timer 42
7.4.1 Requirements 42

7.5 Register map layout 43
7.6 JEITA application 45

8 Verification **46**
8.1 Simulation 46
8.1.1 Normal function 47
8.1.2 Analog errors 48
8.1.3 JEITA application 49
8.1.4 Measurement Error 50
8.1.5 Timer Error 51
8.2 PSL 52
8.3 Coverage 53
8.4 Formal verification 55
8.4.1 SuperLint 55

9 Synthesis **57**

10 Final IP block **61**

Conclusion **62**

Bibliography **63**

Symbols and abbreviations **67**

List of appendices **68**

A Content of the electronic appendix **69**

B SuperLint setting **70**

C Synthesis Data **72**

List of Figures

1.1 Basic working principle of lithium-ion battery [3]. 16
1.2 Lithium-ions battery charging process [10]. 19
2.1 Internal structure of EDLC [13]. 20
2.2 EDLC charging in constant voltage mode. 22
2.3 EDLC charging in constant current mode. 22
3.1 Temperature regions for charging current and voltage [16]. 24
4.1 Design part of VLSI process. 25
4.2 Block diagram of Moore type FSM [21]. 27
4.3 Block scheme of a counter. 29
6.1 Simplified scheme of PQ phase measurement. 32
6.2 Simplified schematic of SAR, analogue part. 33
7.1 Simplified diagram of the whole chip. 35
7.2 Block schematic of top module. 36
7.3 Simplified diagram of the main FSM. 38
7.4 Example waveform for normal charging process. 38
7.5 Block schematic of measurement module. 40
7.6 Simplified diagram of measurement FSM. 41
7.7 Block schematic of timer. 42
8.1 Collected code coverage metrics and their grades for the hardware charger. 53
8.2 Charger FSM states from coverage database 54
8.3 Measurement FSM states from coverage database 54
8.4 Example of error reported by Jasper. 56
9.1 Clock gating for timeout timer. 57

List of Tables

6.1 Input signals from the analog part. 34
6.2 Output signals to analogue part. 34
7.1 Transition states for charger FSM. 37
7.2 Register map layout. 44
8.1 Result of SuperLint. 55
9.1 Area size data from synthesis 58

Listings

8.1 Version of Xcelium (xrun) used. 46
8.2 Underflow propery of timer. 52
8.3 Propery checking if value was measured by SAR. 52
8.4 Underflow propery of timer. 52
8.5 Version of Integrated Metrics Center used. 53
8.6 Version of Jasper Gold used. 55
9.1 Version of Design Compiler (dc_shell) used. 57
9.2 State coding for Charger FSM 58
9.3 Timing report from charger timer. 59
9.4 State coding for Measurement FSM 60

Introduction

Batteries and supercapacitors are essential components in nearly all modern devices. The lithium polymer battery (Li-pol) is one of the most common types, recognized for its excellent power-to-weight ratio. However, charging these batteries poses specific challenges. Lithium polymer batteries require a more careful approach than older technologies, such as nickel-cadmium (NiCd), which have simple charging processes. Ensuring safe and reliable charging is critical; failing to follow proper protocols can lead to battery damage or destruction. Due to the chemical elements used in lithium polymer batteries, explosions or fires can occur, which is why devices utilizing these batteries must have integrated chargers that monitor voltage and temperature during charging.

This thesis is dedicated to exploring the charging process of lithium polymer batteries and EDLCs and creating a comprehensive hardware charger design. The analog components will not be modified, allowing the final design to serve as a drop-in replacement for the previous SW based version or for any application that does not have a CPU. The charger should detect when a battery is connected and commence charging only when supply and battery voltage and temperature are within the limits specified in the register map. Additionally, the charger must comply with JEITA guidelines, which define four temperature zones for all charging phases.

The hardware design will be organized into multiple sub-blocks, each with requirements derived from the original software charger. Once completed, the design will be subjected to industry-standard verification processes. Finally, synthesis will be performed to estimate the area of the charger, providing valuable insights for future projects that may require an integrated charger.

1 Lithium-ion batteries

Lithium-ion batteries are one of the most popular types of rechargeable batteries. The commercialization of lithium-ion batteries started in 1991 in Japan; improvements have been made every year since then. Two primary reasons for the widespread use of lithium-ion batteries are their excellent performance and capacity range. Also, their introduction at the start of the growing consumer electronic market made them popular [1].

1.1 Construction and Materials

Lithium-ion batteries are constructed from a positive electrode, a negative electrode, and liquid electrolytes. Lithium polymer batteries have a composition similar to lithium-ion batteries but have a solid or gel-like electrolyte. The primary chemical principle of lithium batteries is the movement of lithium ions from the negative electrode (anode) to the positive electrode (cathode) through the electrolyte. During charging, the positive electrode releases lithium ions to the negative electrode, and this process is reversed when discharging. Figure 1.1 shows the direction of the lithium ions through the electrolyte during the charging process [1, 2].

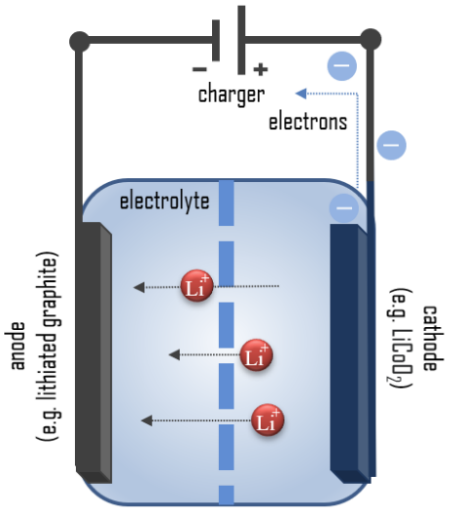


Fig. 1.1: Basic working principle of lithium-ion battery [3].

The material used for negative electrodes needs to store and release Li ions and, more importantly, prevent the creation of lithium dendrites. Dendrites reduce anodes' ability to store and release Li ions, which can cause short circuits, which can cause a chain reaction that starts a thermal runaway. Thermal runaway can cause a fire or even an explosion. Most anode materials, such as graphite, are carbon-based.

Graphite was and still is one of the most widely used anode materials. It offers good mechanical stability and electrical conductivity, can easily store and release Li ions, and, most importantly, has abundant availability and low cost. In recent years, other materials have been tested, such as lithium titanate as an anode. Lithium titanate ($Li_4Ti_5O_{12}$) offers superior thermal stability, higher volumetric capacity, and a longer life cycle than carbon material. However, they also have lower cell voltage and lower capacity. Lithium titanate is used in applications where a longer life cycle is needed. Other materials are available, such as Nb_2O_5, MgH_2, and many others [1, 4].

Cathode materials provide a source of lithium ions for the cell. The original material used was lithium cobaltate $LiCoO_2$, which provides high theoretical capacity, low self-discharge, and high cell voltage but has the lowest thermal stability of any materials used as cathode and high cost. Low thermal stability can cause thermal runaway, which can start a fire; in 2013, a lithium-ion battery experienced a thermal runaway onboard two 787 Dreamliners, causing the grounding of all 787 planes [5]. Substitute derivatives such as $LiNiO_2$ were developed, but they proved more unstable than $LiCoO_2$. Today, one of the most widely used materials as a cathode is $LiFePO_4$, mainly because of its low cost [1, 4].

The liquid electrolyte has many safety issues, mainly because the electrolyte is made from organic solvents. Lithium polymer batteries replace the liquid with polymer or gel-like substances. Polymer electrolytes can be divided into three groups. Solvent-free polymer electrolytes (SPE) consist of polymer matrices and Li salts without the addition of liquid solvents as plasticizers. SPE should fulfill these essential criteria: salt dissociation, allowing movement of ions, dielectric constants, effective separation of charge carriers, and mechanical strength. SPEs are divided based on the type of matrix used. Materials used for the SPE matrix are polyethylene glycol (PEG), polyvinylidene fluoride (PVDF), and many others. Gel polymer electrolyte (GPE) is an intermediate between liquid and SPE. They have improved ionic conductivity as SPE. Polymethylmethacrylate (PMMA), PVDF, propylene carbonate, and others have been commonly used as polymer matrices. Composite polymer electrolytes are created by adding fillers to SPE to increase ionic conductivity and mechanical strength. Examples of fillers are $BaTiO_3$, $PbTiO_3$, $LiNbO_3$, and others [6].

A single battery is commonly referred to as a "cell". A battery is constructed from one or more cells connected in parallel or series; this is done to increase the current or voltage of the battery. A combination of serial-parallel cell connections can be used for specific applications [7].

1.2 Charging process

One of the biggest disadvantages of lithium polymer (or lithium-ion) batteries is the charging process. Lithium polymer batteries cannot be directly connected to a power source; charging current and voltage must be regulated to prevent the battery from overheating, which can cause thermal runaway, fire, or even an explosion. Every cell has defined operating ranges for voltage, current, and temperature. The single cell operating voltage in the range of 2.5 V to 4.2 V, the current is usually expressed by C; 1 C is defined as the constant charging current that will fully charge the battery in one hour. During the charging process, the voltage and, most importantly, the temperature must be monitored to stop the charging process whenever these parameters get out of the allowed range. [8].

The process is divided into three phases, as can be seen in the example figure 1.2:

- **Prequel** (or precharge, trickle current), in this phase, the battery voltage is lower than V_{L}, (usually $V_{\mathrm{L}} \approx 2.5\,\mathrm{V}$, depending on the manufacturer and type). The charging current must be at maximum of 0.1 C, to prevent a large inrush current from the charger reaching the depleted battery. This phase is also used to warm up the battery at the temperature range limit.
- **Constant Current**, battery voltage is between V_{L} and V_{H}. A higher charging current can be applied at 1 C. Most of the battery capacity is reached in this phase.
- **Constant Voltage**, when the battery reaches a voltage higher than V_{H}, the charger acts as a constant voltage source and current drops as can be seen in figure 1.2. When the current drops lower than the "cut-off" current, typically lower than 0.02 C, the power source is disconnected and the process is finished.

Every phase also has a defined timeout and the duration it should take. If the charging process stays in one phase longer than defined, charging needs to be stopped. A prolonged battery charging can indicate an internal fault; by continuing the charging process, destruction of the battery can occur [8, 9].

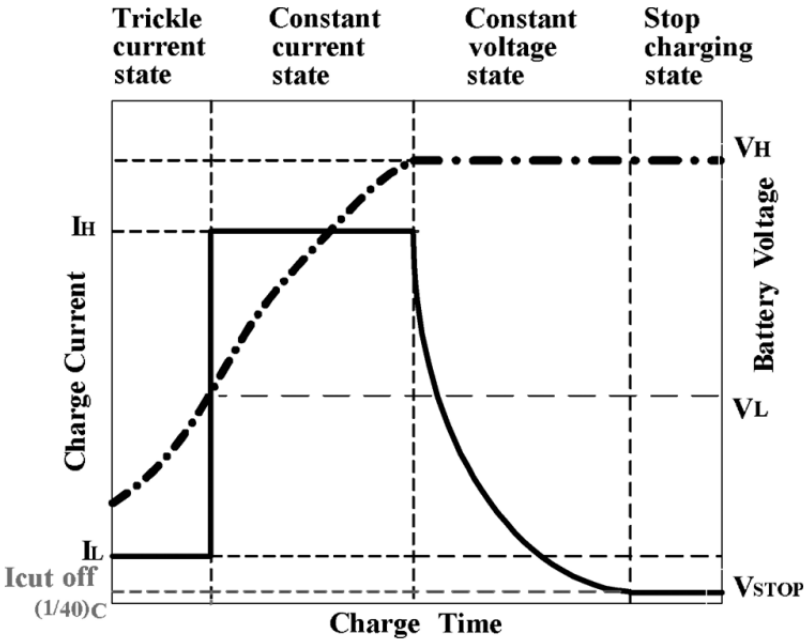


Fig. 1.2: Lithium-ions battery charging process [10].

2 EDLC

EDLC (Electric Double Layer Capacitor) is one type of supercapacitor, functioning similarly to conventional capacitors; they are passive devices that store electrical energy in the field between two plates of opposite charge. EDLCs accumulate charge at the interface between the conductor surface and the electrolyte solution, and conventional capacitors accumulate charge on the two armatures. Compared to lithium polymer batteries, EDLC has a longer lifetime, higher charging and discharging efficiency, lower internal resistance, and smaller energy density [11, 12].

2.1 Construction and Materials

EDLC has a symmetric construction, consisting of two electrodes separated by a membrane, to prevent electrical contact between electrodes, as shown in figure 2.1. An electrical double layer is formed at each electrode and electrolyte interface. This layer was described by H. von Helmholtz in 1853, often referred to as the Helmholtz double layer. The layer is composed of the inner and outer layers [11].
The material from which electrodes are made plays an important role in determining the electrical properties of an EDLC. The surface area of the Helmholtz double layer created on the electrode directly influences the amount of charge that can be accumulated. One of the most used materials is carbon, but other materials, such as metal oxides or polymer conductors, can be used. Carbon has some of the best properties for a good electrode material: low electrical resistance, large surface area per gram ($\approx$ 3000 m^2g^{-1}), mechanical and chemical stability, and it is a low-cost and available material [11].

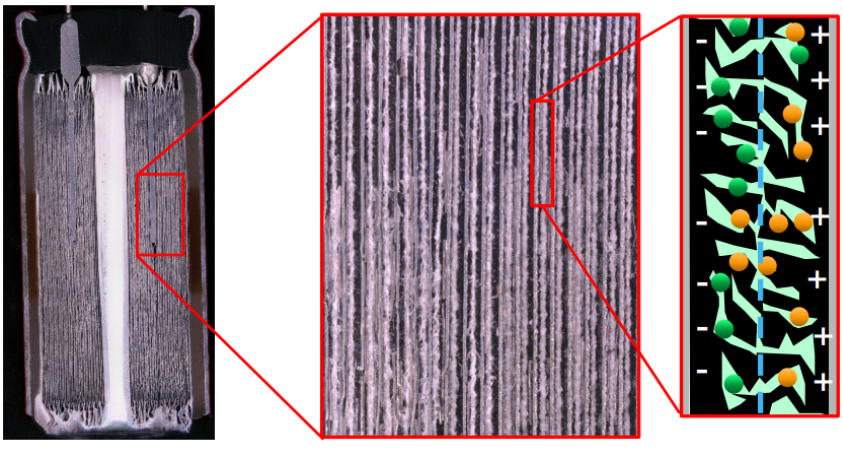


Fig. 2.1: Internal structure of EDLC [13].

As an electrolyte material, three main types are commonly used: organic, ionic, and aqueous. The type of material used determines the safety and performance of EDLC. The main difference is the maximum voltage and ionic conductivity. One of the commonly used organic materials is prepared by dissolving salts of tetraethylammonium tetrafluoroborate ($TEABF_4$) and acetonitrile (AN). The operating voltage of this solution is up to $2.3\,\mathrm{V} - 2.7\,\mathrm{V}$, even at $-40\,^{\circ}\mathrm{C}$. The viscosity of the electrolyte governs the ionic conductivity; the higher the viscosity, the lower the ionic conductivity. Ionic-type electrolytes improve the abilities of organic types, mainly with a nominal voltage higher than $3\,\mathrm{V}$. They have low toxicity, nonflammability, and good stability. They are composed of cations and anions, which determine the electrical properties. Commonly used materials are hexafluorophosphate ($PF_6{}^-$) or tetrafluoroborate ($BF_4{}^-$) and many others. An aqueous type of electrolyte (water-based) is used mainly because of its low cost and high ionic conductivity. However, the operating voltage is around $1\,\mathrm{V}$. They perform poorly at temperatures below $0\,^{\circ}\mathrm{C}$. Typically used materials are sulfuric acid (H_2SO_4) or potassium hydroxide (KOH) [14].

2.2 Charging process

The charging process for the EDLC is simpler than for lithium polymer batteries. There are no phases, and the charging process takes less time, typically seconds or minutes [15].
Two main methods can be used:

- **Constant Voltage** (CV), for power supplies that cannot handle capacitive loads. The voltage of the power supply remains constant, and a protective resistor limits the current; the value of the protective resistor is dependent on the capacity. The voltage of EDLC increases logarithmically and the current decreases exponentially, as seen in the figure 2.2. The charging is stopped when the voltage of the EDLC reaches the maximal value [15].
- **Constant Current** (CC), current is held constant during the charging process. The EDLC voltage increases linearly, as seen in figure 2.3, when the voltage of the EDLC reaches maximum, the power source switches to constant voltage, and the current drops slowly to 0, and charging is stopped [15].

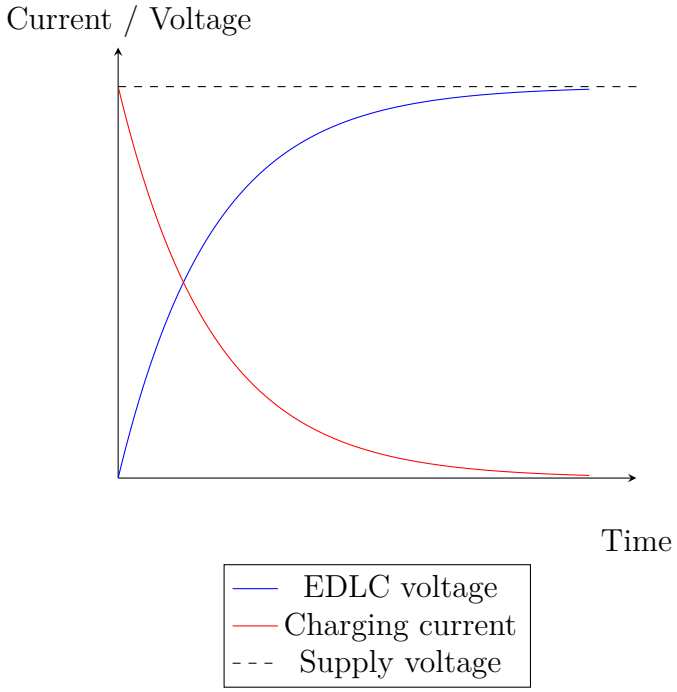


Fig. 2.2: EDLC charging in constant voltage mode.

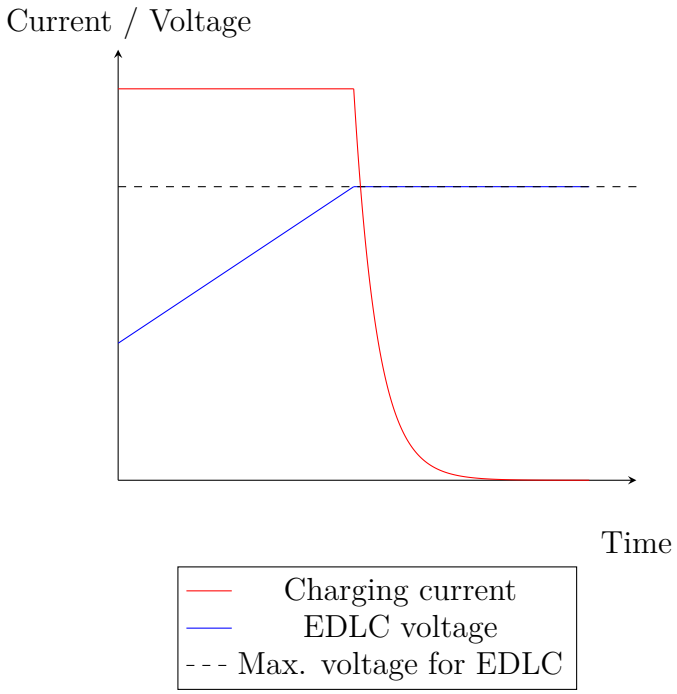


Fig. 2.3: EDLC charging in constant current mode.

3 JEITA

The Japan Electronics and Information Technology Industries Association 2007 created a guide for safe charging secondary lithium-ion cells. The guide describes the basic parameters and temperature ranges for the charging and discharging lithium-ion batteries.

The below limits are important for the charging process [9].

- **Maximum charge voltage** – If the battery exceeds the maximum allowed voltage, excess lithium ions are released from the cathode, which degrades the crystalline structure. This increases the probability of thermal runaway if other problems, such as internal shorts, occur.
- **Standard temperature range** (T2–T3) – Defined as the cell surface temperature range when maximum current and voltage can be used. The typical range for most cells is from 10 °C to 45 °C. This can differ depending on the manufacturer, model, or material type used in cells.
- **Upper-temperature range** (T3–T4) – In this range, a higher quantity of lithium ions migrate from the cathode, lowering the stability of the crystalline structure and decreasing battery safety. Typically from 45 °C to 60 °C. Charging can still be performed, but with reduced maximum values for the voltage and current.
- **Lower-temperature range** (T1–T2) – The rate of migration and reception of lithium ions at the anode is lower. Metallic lithium starts to form on the surface of the anode. The thermal stability of the cell decreases, and the probability of thermal runaway increases. Typically from 0 °C to 10 °C. Charging can be done in this range, but the maximum current and voltage must be reduced.
- **Outside of the temperature range** – Charging must be stopped if the surface temperature exceeds the T1–T4 range. Otherwise, permanent damage or destruction can occur.

The cell manufacturer should provide, in technical documentation, the exact value of how much the voltage and current levels shall be decreased in the lower and upper temperature ranges. However, most of them provide only information about the standard temperature range. Figure 3.1 directly applies the Texas Instruments JEITA guideline. It shows what modifications in lower and upper temperature ranges are done, but this is not universal, even for all Texas Instruments chargers; some chargers have programmable current and voltage in all phases, while others have only in the lower range. The guideline also defines the operational range (T5–T6), also known as the center range, smaller than the standard range [9].

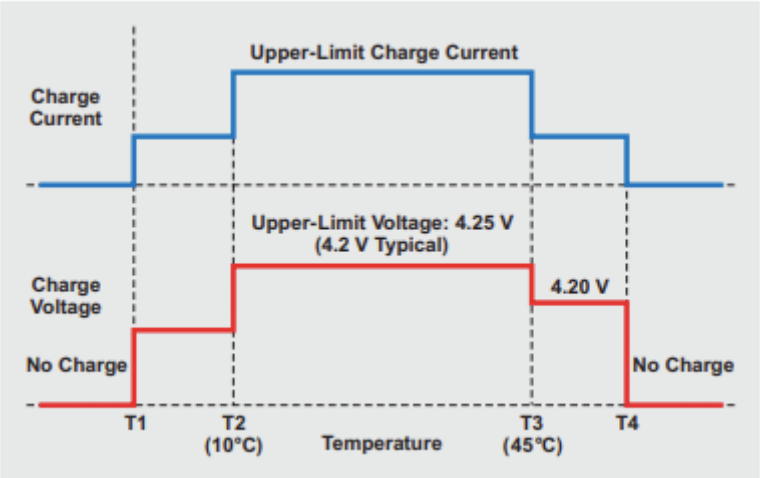


Fig. 3.1: Temperature regions for charging current and voltage [16].

4 Digital design hierarchy and principles

Digital design is usually composed of connected blocks or structures. These blocks are created and independently tested to verify their function. Usually, the isolated blocks are designed to be parametric in their key features (i. e., address and data bus widths, number of counter stages, ...) so that they can be instantiated multiple times in the design if needed, just by changing their parameters. Figure 4.1 describes the iterative steps done in this thesis to create and verify the digital part of the hardware charger [18].

The following section describes the basic building blocks needed to implement a hardware charger.

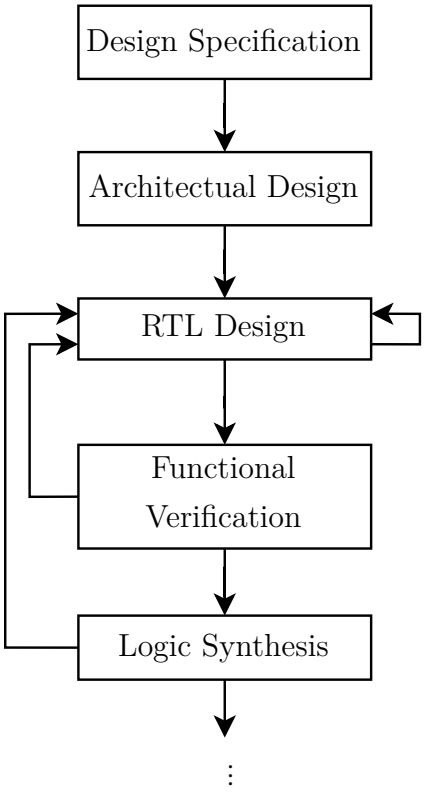


Fig. 4.1: Design part of VLSI process.

4.1 FSM

The finite state machine provides an abstraction for a sequential circuit with a defined operation state. The "problem" that FSM is solving is divided into finite amounts of states, each of which has a defined output value and a definition of the next state. In synchronous logic implementation the transitions from the present

state to the next state occur by triggering the edge of the clock signal, typically by the rising edge. Formally, FSM can be defined as a mathematical model specified by six entities: *set of states*, *starting state*, *input signals*, *output signals*, *next state function*, and *output function* [17, 18].

There are two types of FSM: Moore and Mealy. Both can describe the same problem; the main difference is in the output function. In Moore-type FSM, the states of the outputs are given only by the current state, as can be seen in figure 4.2. This makes output signals synchronous to clock transitions, which prevents the probability of glitches at the output. Mealy-type FSM outputs are decided by inputs and current state; this decreases the reaction time of outputs compared to the Moore-type. The Mealy-type has a direct combinational path from input to output, which can create glitches at the output. For example, the Mealy-type is preferred when a quick reaction is required in signal edge detection [18].

The second difference is that the Mealy-type FSM usually has fewer states than the equivalent Moore-type FSM. Theoretically, in computer science, Moore and Mealy types are equivalent; any Mealy-type can be transformed into an equivalent Moore-type and vice versa. In practice, some Moore-types can be transformed into Mealy-types, but not in another way [17, 18, 19].

Different state encoding schemes utilize varying numbers of bits to represent states, which can lead to the presence of unused states [20, 21].
Commonly used types of encoding:

- **Binary** coded states according to the binary sequence, using a minimal number of bits. However, the transition of one or more bits may occur in parallel between neighbouring states. The number of bits needed can be calculated using the 4.1 formula, and the number of unused states can be calculated using 4.2 [18].
- **Gray** assigned states according to the Gray sequence. Bit compactness is the same as binary, but only one bit changes at the transition between neighbouring states. This reduces the chance of unwanted state transitions. The number of bits needed, and the number of states unused for Gray encoding, can be calculated with equations 4.1 and 4.2 [18].
- **One-hot** assigns one bit to each state, for n state FSM, the n-bit register is required. Next state combinatory logic is minimized. The one-hot encoding has a variation called "Almost One-hot"; it has a special reset (or default) state with the value of all zeros. This reduces the number of bits of encoding needed by one. The number of unused states for one hot can be calculated with the formula 4.2, for "Almost One-hot" is one less unused state [18].

$$n = \lceil \log_2 m \rceil \tag{4.1}$$

$$k = 2^n - m \tag{4.2}$$

Where: m is the number of used states, n is the number of bits, and k is the number of unused states [18].

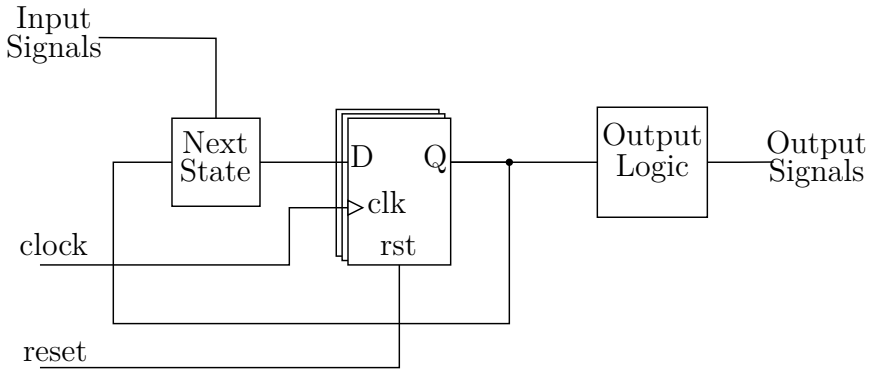


Fig. 4.2: Block diagram of Moore type FSM [21].

4.2 Register map

Often, small memory is required in design, but even the smallest types of memory are larger (in terms of byte size) than register map. Also, very often a reduced read and write access time is required into such a small memory. An effective solution is to combining multiple registers into a larger system known as a register file or map. Registers within this map can have different access modes. Additionally, a register map can be assigned to a specific address space in the design, allowing software (or hardware) managing the design to read or write values from the map [18].

A register is constructed from multiple D flip-flops (DFF) and is a fundamental building block in almost every digital application. The types of information they typically hold include data, addresses, configuration, or status bits. Registers fall under the category of sequential logic and are built using combinatorial logic, as illustrated in the relevant figure. Typically, they feature separate input and output ports and may include a clock signal. Registers can be configured to respond to the rising or falling edge of the clock signal, often incorporating an enable signal[20, 22].

4.2.1 D-FF based resynchronization

Input signals for the analog or digital side do not always have to be synchronous with the clock signal and often have glitches. This is why input signals must be synchronized before they can be used in digital logic. The simple two-bit shift

register synchronizes the input signal, which is delayed by one or two clock cycles. Note that the delay in the synchronizer is not deterministic [17, 18].

Also, it is important to have the input of the synchronizer is glitch-free, which can be achieved by generating its input signal from the register, not from combinatorial logic [17, 18].

4.3 Counters

Counters are among the most fundamental blocks in digital design. They are used in many typical applications: timers, memory address generators, program counters, frequency dividers, signal processing or generation, and many more [23].

A counter can be described as a simple finite state machine that circulates its internal state through a set of states, with a decoder at the output. Counter state encoding is one of the most important things to consider when designing a counter for a specific application. Encoding determines almost all properties of the counter: size, maximum frequency, number of bits that change between each state (Hamming distance), power consumption, whether an output decoder is required, and more. There are two main types of counters: synchronous (all registers change their state simultaneously) and asynchronous (also known as a ripple counter). There exist many different implementations of counters such as *binary*, *Johnson*, *Gray*, *LFSR*, and others [18, 23].

In the figure 4.3, the block diagram of the binary synchronous decrementing counter can be seen. It is divided into three parts:

1. **Next State**, the size and complexity of **next state** depends on the type of encoding. In case of figure 4.3, three things can occur. The next state will not change; the value will be decremented by one, or a new value will be loaded into registers.
2. **Register**, D-type flip-flop that stores the current value. Equation 4.1 calculates the count of D-FFs (registers) needed.
3. **Decoder**, it depends on the use case, sometimes the output decoder is unnecessary.

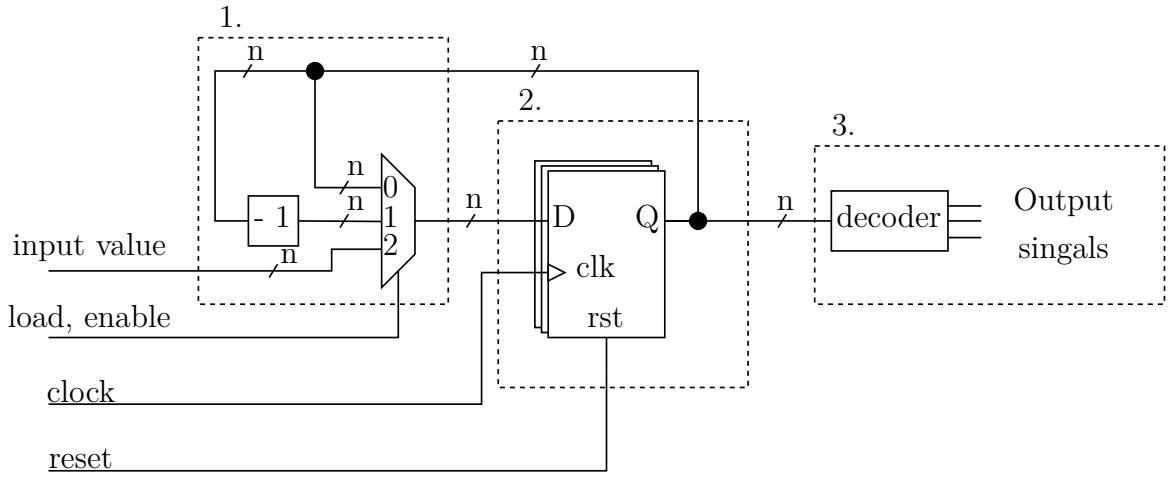


Fig. 4.3: Block scheme of a counter.

4.4 Verification

The complexity of digital design increases every year, currently verification costs have reached around 60 % of the whole development. Two approaches can be taken for verification: simulation or formal-based verification. Typically, both methods are used to verify the design, mainly because they complement each other [24].

Every design has a specification from the customer that defines the main functionality of the design. From this specification, individual blocks and their connections are created, but more importantly, requirements for the whole design and individual blocks are defined. Requirements are simple statements that describe how the design or individual parts function. Often, a single requirement has multiple subrequirements that more precisely specify the function. In addition, the requirement coverage serves as an indication of how much the design functionality was verified. Requirements can be verified in multiple ways. With the help of PSL (Property Specification Language) or SVA (System Verilog Assertions), requirements can be written as properties and assertions directly in the design and verified during simulation [25].

4.4.1 Simulation

Simulation uses a testbench to drive the inputs to the DUT (device under test); the output signals from the DUT are compared with the DUT reference model. Multiple ways of setting inputs can be used:

- **Direct testing** – Tests (scenarios) are manually created by the verification engineer; they have control over the individual test, but need a deep knowledge

of the design and specification. The manual test can target corner cases and important parts of the design, but requires time to be set up [25, 26].

- **Constrained random testing** – The inputs for the DUT and the model are generated randomly, but in realistic ranges, so the inputs of real life are only set and not values that cannot occur in a real device [25, 26].

One piece of information that simulation of tests provides, other than whether the test passed or failed, is a code coverage. Code coverage provides information about which lines of simulated code were triggered. Coverage from multiple tests is merged because not every test has to trigger every line of code. However, more important information is what lines were not triggered; this can indicate unreachable or unused parts of the code. The ideal coverage is 100 % [25, 26].

4.4.2 Formal

Formal-based verification uses a different strategy than simulation-based verification; it does not use the testbench to drive inputs and compare outputs, rather it uses assertions and properties, and tries to mathematically "prove their correctness". Typically, two methods are used:

- **Equivalence checking** verifying that two designs are equivalent [26].
- **Model checking** verifies that the design satisfies the mathematical model created from the specification and requirements [26].

Lint performs structural control of the created RTL (Register Transfer Level), such as coding style, simulation-synthesis mismatch, race conditions, FSM reachability, state transitions, and much more [27].

Many large designs today use multiple clocks with different periods to create a more power-efficient design. The CDC (Clock Domain Crossing) checks signals between domains and controls more functions, such as synchronization, convergence, and many more. The same tool is also used for reset signals, because many criteria that apply to the CDC also apply to the RDC (Reset Domain Crossing) [28].

5 Tasks for bachelor's thesis

Analyze how the original charger was implemented and how the analog part is implemented. Create a new digital block-level architecture with the same function and features as the original SW based charger and implement the JEITA guidelines.

The hardware implementation will be split into multiple blocks; write their requirements.

The design shall meet the following criteria:

1. Implement the same behaviour as the same as the original SW based charger
   - Follow the same charging phases
   - Measure cell or EDLC voltage, temperature, and source voltage for JEITA compliance
2. All inputs have to be synchronized
3. All outputs have to be glitch-free
4. All requirements shall be checked by verification
   - Create scenarios to verify primary functions
   - Use available tools and applications such as Lint for FSM reachability and Coverage for dead code elimination
5. Run synthesis to check the RTL code is synthesizable
   - Extract data from the report about STA violation, approximate size, and FSM encoding

6 Optimization target

The original charger was a combination of hardware and software. The primary functions of the hardware part were to set outputs for the analogue part and to store input values from the analogue part in the register map. The software was responsible for the entire charging process, reading values from the register map about the current phase of charging, setting correct limits for current, measuring voltage and temperature using SAR (Successive approximation register), and controlling time in each phase.

6.1 Analogue part

The analogue side is not the central part of this thesis; this section explains what signals are input and output, and their function. The analogue part of the charger consists of two main blocks: a set of comparators that compare cell or EDLC voltage to the reference value and a current-limiting circuit.

6.1.1 Charging

It consists of two parts: comparators that compare cell (or EDLC) voltage to a reference value and a current-limiting circuit.

The signals of the comparators are described in table 6.1; from a digital part perspective, they are asynchronous, so they need to be synchronized with the clock signal. Figure 6.1 is an example circuit schematic that compares the battery voltage with a reference value. All other signals in table 6.1 are constructed similarly.

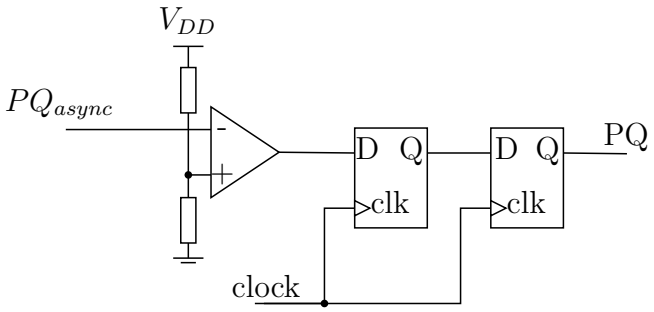


Fig. 6.1: Simplified scheme of PQ phase measurement.

6.1.2 JEITA related measurement

Inputs (PQ, CC, CV) described in table 6.1 only provide information about charging phase changes, not the actual value of the cell voltage (or EDLC). Temperature

measurement is required to be JEITA compliant. Cell (or EDLC) voltage and temperature change slowly so that measurement can occur over multiple clock cycles without significant error. Any measurement circuit can be used; the circuit described in the following is a simple SAR, constructed from a comparator and DAC. Temperature is measured as voltage drops on components, such as NTC or PTC thermistors.

ADC SAR block implements successive approximation conversion controlled by a digital part. The analogue side of SAR sets the reference value to the comparator (table 6.2, pin SAR_DAC), and the digital side stores sample information; if the measured value on input (table 6.1, pin SAR_input) is higher or lower than a reference voltage. In this case, the reference value control register is 6 bits, so the measurement takes six clock cycles.

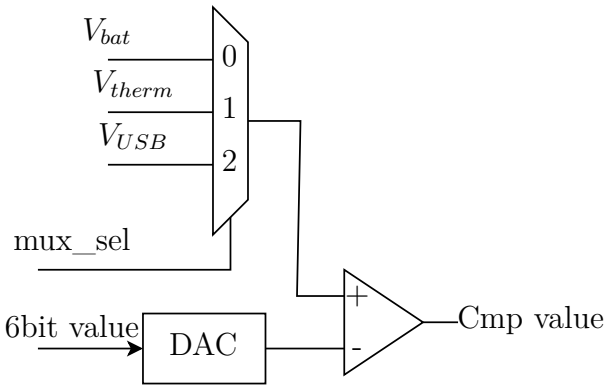


Fig. 6.2: Simplified schematic of SAR, analogue part.

6.1.3 Digital part interface

All status signals from the analogue part, described in table 6.1, are 1 bit wide and synchronized with the corresponding clock signal. The output signals are generated from the register map to create a glitch-free signal, synchronized with the clock. The width and description of each output are defined in table 6.2.

Table 6.1: Input signals from the analog part.

Name	Description
Ready	Indicate battery is connected
OVD	Detected over voltage
UTD	Detected under temperature
OTD	Detected over temperature
PQ	Charger is in the prequel phase
CC	Charger is in the constant current phase
CV	Charger is in the constant voltage phase
Done	Charging finish
SAR_input	Comparator value form SAR

Table 6.2: Output signals to analogue part.

Name	Width	Description
SAR_mux	1	Selector for SAR multiplexer
SAR_DAC	6	Reference value for SAR
I_value	4	Set max. current value

6.2 Original software

In the original solution, the SW controlled the measurement and phase transitions. The signals from 6.1, which indicate phase transitions, were sampled in a single register. The software then matched the value of this register to predefined values and set the correct values for the output. Incorrect state transitions were considered as an error, and the charging was stopped.

7 New block level architecture

The digital side is the central part of this thesis. Newly, it shall be responsible for measuring the voltage and temperature, monitoring the proper duration of each particular phase of the charging process, and setting a correct current limit. The software can still control the changing process, but in this implementation, its primary function is to monitor hardware functions only and step in only if the hardware does not handle the problem correctly.

The hardware charger is integrated into the larger SoC. A simplified schematic can be seen in figure 7.1. For loading values to register map two IP blocks are used in the final design: the NVM controller and the memory model. Both are from different projects and were reused and modified for this thesis. A small FSM was implemented to handle the loading values from NVM to RM, start CRC calculation (it is built into the NVM controller), and compare the results.

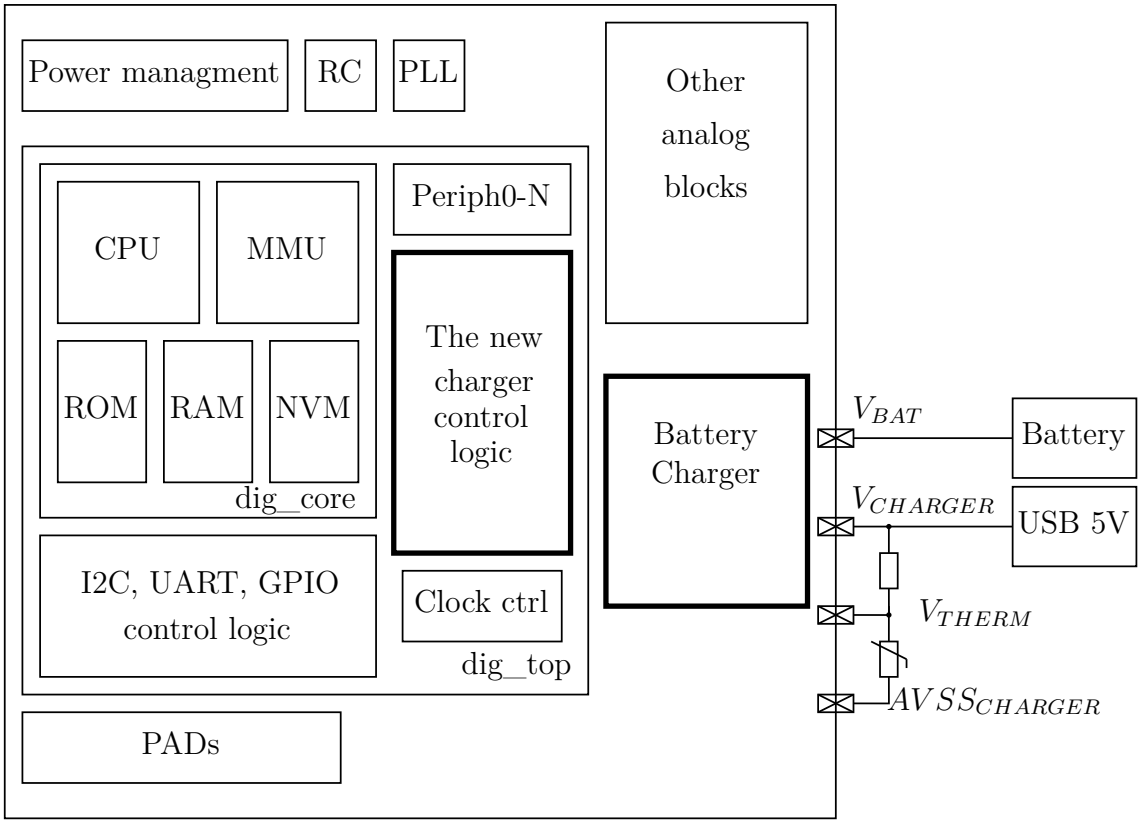


Fig. 7.1: Simplified diagram of the whole chip.

7.1 Charger top Level

The main function of the top-level charger block is connecting the individual sub-blocks to the analogue part. The charger top-level block is composed of the following blocks: register map, FSM, Measurement, and Timer. The design is divided into blocks as shown in figure 7.2.

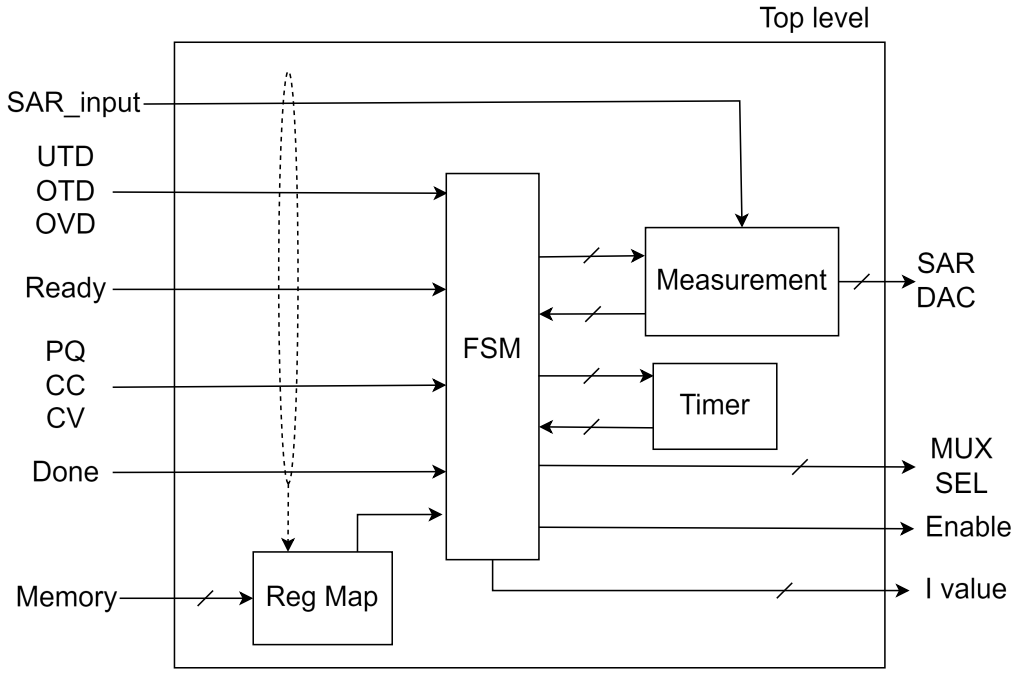


Fig. 7.2: Block schematic of top module.

7.1.1 Requirements

Req.7.1.1 The hardware design shall be implemented as shown in figure 7.2

Req.7.1.2 RM shall have layout as shown in table 7.2

Req.7.1.3 Charging process shall start only after the RM is loaded from external and passes CRC sum comparison

Req.7.1.4 All inputs from the analogue part shall be properly resynchronized

Req.7.1.5 All outputs to analogue are glitch-free and driven from DFF

7.2 Top Level FSM

The FSM at the top level is the main automata that controls the whole charging process. It is responsible for starting and monitoring (e.g., proper transitions between PQ, CC, CV phases) and properly finishing the charging process. It needs to react and correctly handle any changes or errors during the charging process. The state diagram for the main (charger) FSM is shown in figure 7.3, example is shown in figure 7.4.
The states for the main FSM can be divided into three groups:

- Charging disabled
  - IDLE, starting state (power-on reset), the FSM waits until the cell or EDLC is connected
  - LOAD, load data from memory to register map
  - END, the charging has finished
- Charging enabled
  - PQ, start of the prequel phase, period for timer and current limit are loaded
  - Measur. PQ, the timeout timer is started, and measurement is enabled
  - CC, the start of the constant current phase, period for the timer, and current limit are loaded
  - Measur. CC, the timeout timer is started and measurement is enabled
  - CV, the start of the constant voltage phase, period for the timer, and current limit are loaded
  - Measur. CV, the timer is started, and measurement is enabled
- Error, charging must be disabled,
  - FSM can end up in this state from measurement, when the measured value exceeds the given limits, or due to an error from the analogue part

Table 7.1: Transition states for charger FSM.

	Ready	**PQ**	**CC**	**CV**	**Done**
S1	1	1	0	0	0
S2	1	0	1	0	0
S3	1	0	1	1	0
S4	1	0	0	0	1

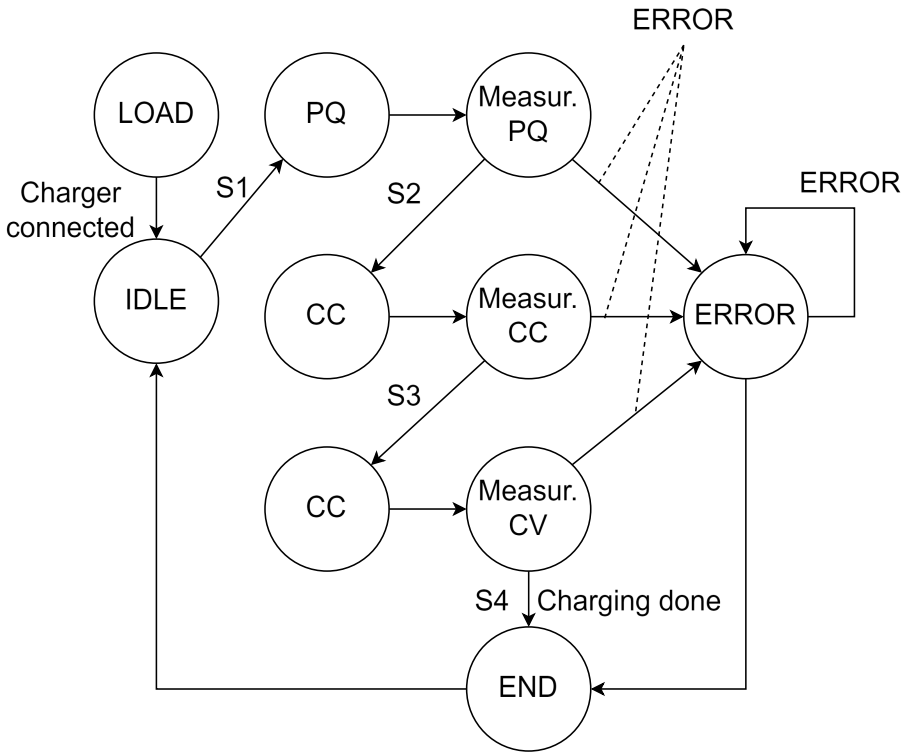


Fig. 7.3: Simplified diagram of the main FSM.

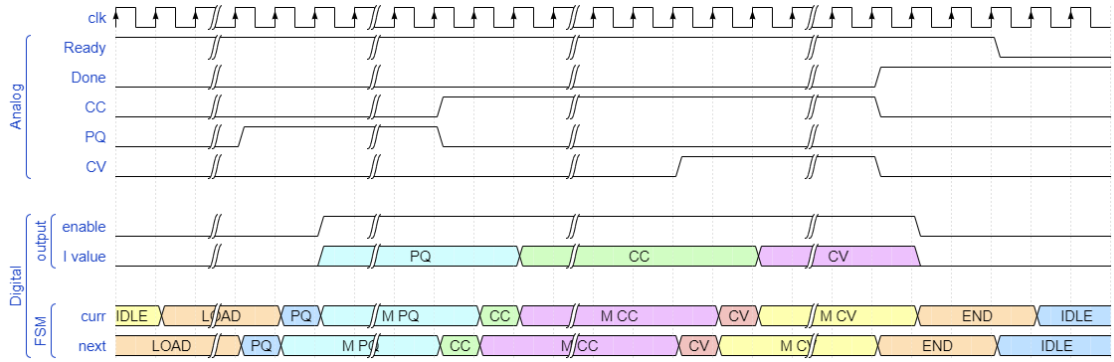


Fig. 7.4: Example waveform for normal charging process.

7.2.1 Requirements

Req.7.2.1 The charger FSM state transition diagram shall follow Fig. 7.3

Req.7.2.1.1 All the state transitions of the original SW solution (defined in 7.1) shall be identically implemented in the new HW Charger FSM

Req.7.2.1.2 Any other combination shall be treated as an error

Req.7.2.1.3 State transition shall happen only after a proper measurement is done

Req.7.2.2 Charging must be stopped if there is an error from the measurement or the analogue side

Req.7.2.3 During the charging process, only the following signals from the digital side $pml\_bc\_charge\_en$, $pml\_bc\_ifullchg$, and $pml\_bc\_trim\_vref$ can change

Req.7.2.3.1 There is a 300 µs delay between $bias\_ready$ going to '1' and $charger\_det$ going to '1'

Req.7.2.4 Once per second V_{bat}, V_{therm} (temperature), and V_{usb} measurement shall be done

Req.7.2.4.1 Temperature (V_{therm}) and USB (V_{usb}) measurement can be disabled optionally

Req.7.2.5 During every stage of the charging process (PQ, CC, CV), the HW shall control the maximum time for a given stage

Req.7.2.5.1 Timer shall be implemented as countdown timer

Req.7.2.5.2 Timer value reaching zero is considered as an error

Req.7.2.5.2.1 Control logic shall terminate the charging process in such a case

Req.7.2.6 The charger shall implement the JEITA guidelines

Req.7.2.6.1 Based on Vtherm measurement, the timer shall double the timeout value in a temperature range of T_1 – T_2 or T_3 – T_4

Req.7.2.6.2 The output current ($pml\_bc\_ifullchg$) and trim resistor ($pml\_bc\_trim\_vref$) shall be halved, when Vtherm is outside the normal range, as shown in 3.1

7.3 Measurement module

The measurement block implements the digital part of the SAR measurement. Voltage and temperature change slowly, so measurements shall be performed once per second. The limit values (voltage and temperature) for each stage are provided from the top-level module. The measurement block is then subdivided into more blocks, consisting of FSM, timer, and SAR, as shown in figure 7.5.

The secondary FSM shall switch between temperature and voltage measurement and starts the approximation. It will wait until the main FSM is enabled. Firstly, the voltage measurement is started and evaluated, and then the temperature measurement is started and evaluated. If the measured values are within the range given by the limit values, the charging may continue; otherwise, the error is set, and the charging shall be stopped. After measuring the temperature and voltage, it will wait for one second. The time between measurements is set in the register map. The transition of states for the measurement cycle can be seen in figure 7.6.

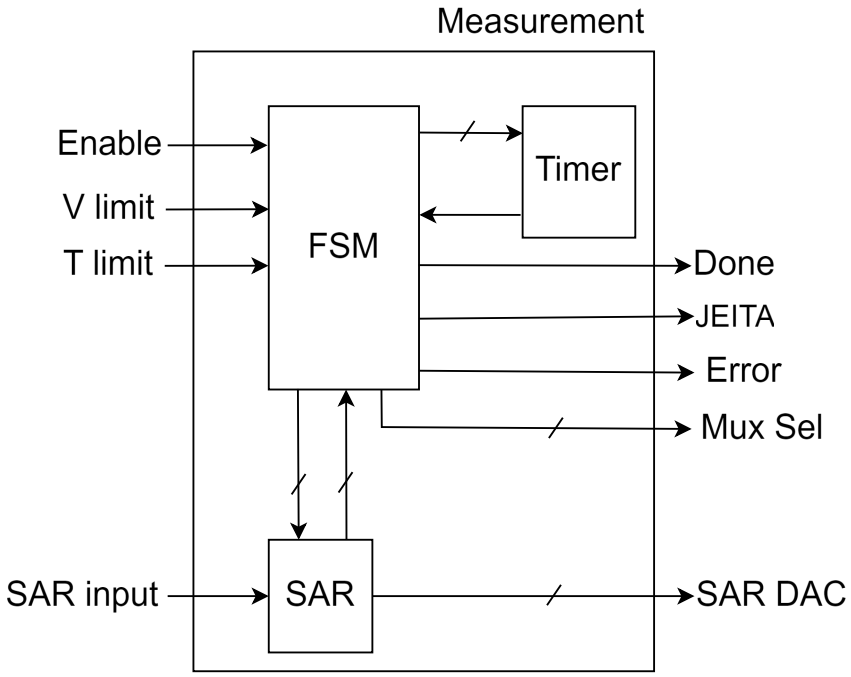


Fig. 7.5: Block schematic of measurement module.

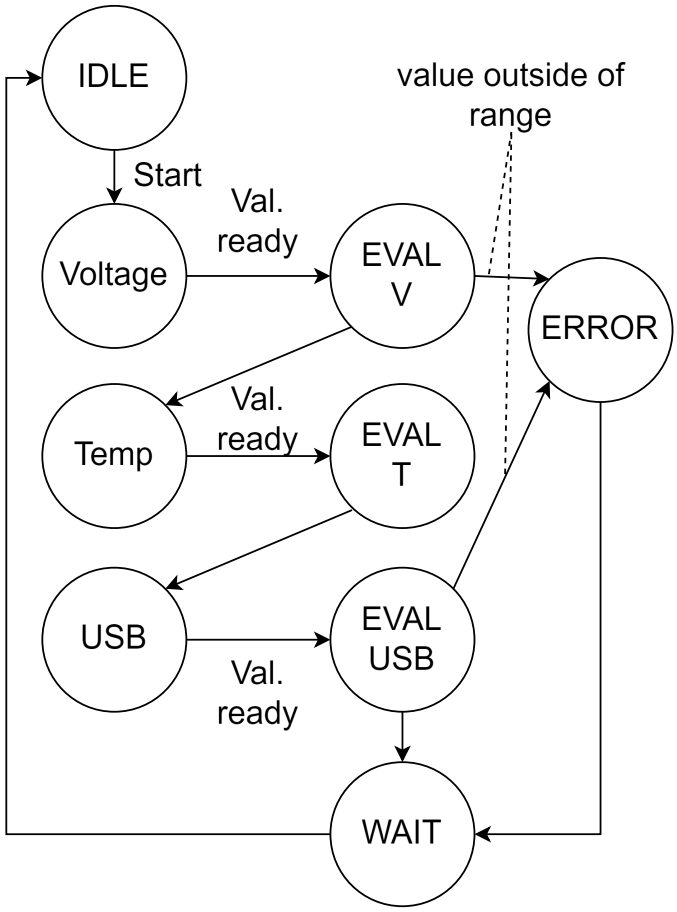


Fig. 7.6: Simplified diagram of measurement FSM.

7.3.1 Requirements

Req.7.3.1 Measurement logic FSM shall follow the figure 7.6

Req.7.3.2 Particular voltage measurement shall implement the SAR algorithm using an existing HW IP

7.4 Timer

Figure 7.7 shows the binary counter. The initial value is loaded into the main register, which is decremented in each clock cycle until it reaches zero. The JEITA guideline recommends doubling the time in each stage if the temperature is outside the normal range. This is achieved by adding a single-bit D-type flip flop that acts as a toggle; the frequency is divided by half.

The timer module is instantiated twice, each instance is able to divide time by half. However, only the timer at the top level, used for measuring the time in each charging stage, will use this feature.

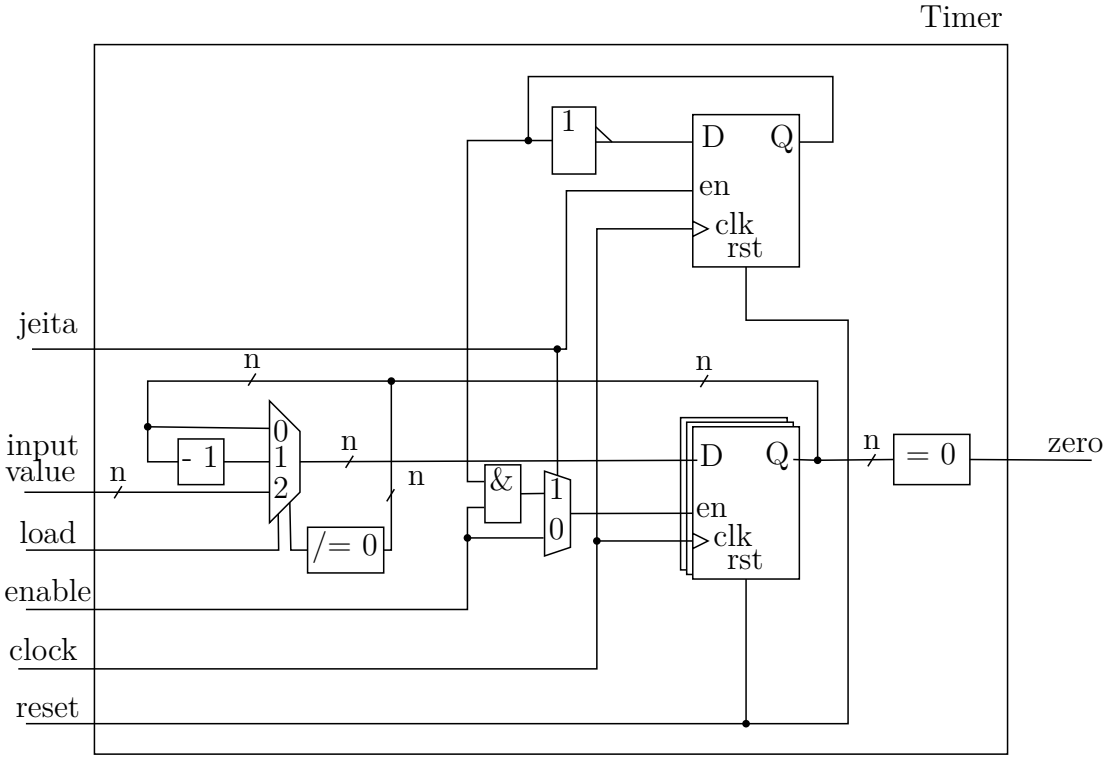


Fig. 7.7: Block schematic of timer.

7.4.1 Requirements

Req.7.4.1 Timer must stop at the value of zero

Req.7.4.1.1 Timer shall not underflow

Req.7.4.2 The timer shall have the ability to double the timeout value

7.5 Register map layout

Register map stores three types of information: the input signals (status flags), the output signals (control), and the intervals (voltage, timing, and temperature intervals). The register map sections follow each other consecutively. Registers can be compacted more together depending on the register's map configuration. Each register is one byte wide.

Tables 7.2 describe the layout of the register map. The first three registers are input signals accessed mainly by the host CPU (software) if the charger is integrated into a larger system. These registers are RO ("Read Only"). The value is written directly to them from the analog side.

The following three registers are output signals. They are RW ("Read and Write"). Both hardware and CPU (software) can write values to them. The output of this register is directly connected to the output signals (table 6.2).

The rest of the registers are for intervals. The type of registers is "Read and Write"; values are written to them from memory only at the beginning of the charging and do not change during the charging process. Only software can override the values in these registers during charging.

Table 7.2: Register map layout.

Address	Type	Width	Name	Description
0x00	RO	8	Flags	Error indication Over voltage Under temperature Over temperature
0x01	RO	8	Analog flags	PQ, CC, CV Ready, Done SAR_Input
0x02	RO	8	SAR	SAR_mux, SAR_DAC
0x03	RW	4	I_val	Current limit value
0x04	RW	8	TimePQ_L	Timer value for PQ LSB
0x05	RW	8	TimePQ_M	Timer value for PQ MSB
0x06	RW	8	TimeCC_L	Timer value for CC LSB
0x07	RW	8	TimeCC_M	Timer value for CC MSB
0x08	RW	8	TimeCV_L	Timer value for CV LSB
0x09	RW	8	TimeCV_M	Timer value for CV MSB
0x0A	RW	8	I_PQ	Current limit for PQ: normal and reduced
0x0B	RW	8	I_CC	Current limit for CC: normal and reduced
0x0C	RW	8	I_CV	Current limit for CV: normal and reduced
0x0D	RW	6	Trim_PQ	Current trim for PQ
0x0E	RW	6	Trim_PQ_J	Current trim for PQ, JEITA
0x0F	RW	6	Trim_CC	Current trim for CC
0x10	RW	6	Trim_CC_J	Current trim for CC, JEITA
0x11	RW	6	Trim_CV	Current trim for CV
0x12	RW	6	Trim_CV_J	Current trim for CV, JEITA
0x13	RW	6	V_min	Minimum voltage limit
0x14	RW	6	V_max_PQ	Maximum voltage limit for PQ
0x15	RW	6	V_max_CC	Maximum voltage limit for CC
0x16	RW	6	V_max_CV	Maximum voltage limit for CV
0x17	RW	6	T_1	T1 temperature limit
0x18	RW	6	T_2	T2 temperature limit
0x19	RW	6	T_3	T3 temperature limit
0x1A	RW	6	T_4	T4 temperature limit

7.6 JEITA application

JEITA (as described in sec. 3) is not defined as a standard that must be strictly followed but is more like a guide or recommendation. The manufacturer of a cell or EDLC usually specifies the normal temperature range for charging in the technical documentation (commonly referred to as a datasheet). Lower and higher temperature ranges are often omitted from technical documentation, so JEITA compliance is done slightly differently in every charger.

The charger design in this thesis supports four temperature zones. In the register map (Table 7.2), four registers T_1 to T_4 define each zone. Each phase has two current values, one for normal and one reduced.
These zones are:

1. Outside of range
    - Lower then T_1 or higher then T_4
    - Charging is disabled and cannot be started
2. Lower range
    - Temperature higher or equal to T_1 and lower than T_2
    - Charging is enabled, but I_Phase_J and Trim_Phase_J value have to be used
    - Timeout has to be doubled
3. Normal range
    - Temperature higher or equal to T_2 and lower than T_3
    - Charging is enabled, normal values are used
4. Higher range
    - Temperature higher or equal to T_3 and lower than T_4
    - Charging is enabled, but I_Phase_J and Trim_Phase_J value have to be used
    - Timeout has to be doubled

8 Verification

The main goal is for the hardware design to perform as well as the original software. This can be achieved in many ways through simulation, assertions, lint, code coverage, function coverage, and more.

8.1 Simulation

The scenarios are designed to cover all the requirements (sec. 7.2.1 and 7.3.1). All tests were run with the Xcelium simulator "xrun", version 8.1.

For simulation, project was connected to SV-RNM model of charger that simulated different batteries, with error simulation and analog part of charger [29].

Selected tests that test main functions of the charger are described below. Before start of every test, all signals from the analog side are set to '0', and the reset line is held at the active value for at least five clock cycles. This starts all tests from the off state (power off). NTC used for measurement has the reference voltage given by the USB voltage for all tests and shall be 5.0 V. All tests will use the same battery type values for charging current, voltage, and temperature limits. The analog part is not present during simulation; signals that behave like the analog part will be generated in tests by the model.

Listing 8.1: Version of Xcelium (xrun) used.

```
>xrun -v
xrun: 24.09-s002: (c) Copyright 1995-2024 Cadence Design
   Systems, Inc. ...
```

8.1.1 Normal function

The normal function without errors is tested. The temperature is set to be in the normal range, not to trigger the JEITA function.

Requirements covered in this scenario: Req.7.2.1, **Req.7.2.1.1**, **Req.7.2.1.3**, **Req.7.2.3**, **Req.7.2.4**, **Req.7.2.5**, **Req.7.3.1**, **Req.7.3.2**

Scenario description:

1. Set the battery voltage at 2.5 V, the temperature is set in the middle of the normal operating range (30 °C).
2. Verify signal Ready is set to '1'.
   - The interval values are loaded into the RM at this point.
   - The charger can be started.
3. Delay for a 100 µs.
4. Verify signal PQ is set to '1'.
   - The charger needs to set the output current to the PQ value and enable the signal.
5. Battery model slowly increased voltage to 3.0 V, with an increase of 10 mV over a time of 200 µs.
6. Verify that PQ is clear and CC is set to '1'.
   - The charger needs to set the output current to the CC value.
7. Battery model increases voltage to 4.15 V, with an increase of 10 mV over a time of 150 µs.
8. Verify CV is set to '1'.
   - The charger needs to set the output current to the CV value.
9. Battery model increases voltage to 4.2 V, with an increase of 10 mV over a time of 100 µs.
10. Verify that CV is clear, and Done is set to '1'.
    - The charger needs to set the output current to 0 and disables charging.

8.1.2 Analog errors

Test controls correct response to errors raised by analog.
Requirements covered in this scenario: Req.7.2.1, **Req.7.2.1.1**, **Req.7.2.1.3**, **Req.7.2.2**, **Req.7.2.3**, **Req.7.2.4**, **Req.7.3.1**, **Req.7.3.2**
Scenario description:

1. Set the battery voltage at 2.5 V, the temperature is set to the middle of the normal operation range (30 °C).
2. Verify signal Ready is set to '1'.
    - The interval values are loaded into the RM at this point.
    - The charger can be started.
3. Delay for 100 µs.
4. Verify signal PQ is set to '1'.
    - The charger needs to set the output current to the PQ value and enable the charging process.
5. Battery model slowly increases voltage to 3.0 V, with an increase of 10 mV over a time of 200 µs.
6. Verify that PQ is clear and CC is set to '1'.
    - The charger needs to set the output current to the CC value.
7. Verify one of the detection signals (OVD, UTD or OTD) is set to '1';
    - The charger needs to stop the charging process immediately and does not continue or restart it unless it is restarted from the CPU.

8.1.3 JEITA application

JEITA is required to lower the charging current and increase the time.
Requirements covered in this scenario: Req.7.2.1, **Req.7.2.1.1**, **Req.7.2.1.3**, **Req.7.2.3**, **Req.7.2.4**, **Req.7.2.5**, **Req.7.2.6**, **Req.7.2.6.1**, **Req.7.2.6.2**, **Req.7.3.1**, **Req.7.3.2**, **Req.7.4.2**
Scenario description:

1. Set the battery voltage at 2.5 V, the temperature is set to the middle of the normal operation range (30 °C).
2. Verify signal Ready is set to '1'.
   - The interval values are loaded into the RM at this point.
   - The charger is started.
3. Delay for 100 µs.
4. Verify signal PQ is set to '1'.
   - The charger needs to set the output current to the PQ value and enable the charging process.
5. Battery model slowly increases voltage to 3.0 V, with an increase of 10 mV over a time of 200,µs.
6. Temperature is increased to 45 °C over time of 200 µs.
   - When the temperature reaches over 40 °C, the remaining allowed time must be doubled and the output current reduced to half.
7. Verify that PQ is clear and CC is set to '1'.
   - The charger needs to set the output current to the CC value. The time is doubled, and the maximum current is reduced to half.
8. Battery model slowly increases voltage to 4.15 V, with an increase of 10 mV over a time of 150 µs.
9. CV is set to '1'.
   - The charger needs to set the output current to the CV value.
10. Battery model slowly increases voltage to 4.2 V, with an increase of 10 mV over a time of 100 µs.
11. Verify that CV is clear, and Charging done is set to '1'.
    - The charger needs to set the output current to 0 and disable charging.

8.1.4 Measurement Error

This test is designed to verify the ability to stop the charging process if the measured voltage is out of range given by the values in RM. Maximum value for the PQ phase set to 2.2 V, so error happen in PQ phase.

Requirements covered in this scenario: Req.7.2.1, **Req.7.2.1.1**, **Req.7.2.1.3**, **Req.7.2.2**, **Req.7.2.4**, **Req.7.3.1**, **Req.7.3.2**

Scenario description:

1. Set the battery voltage at 2.0 V, the temperature is set to the middle of the normal operation range (30 °C).
2. Verify signal Ready is set to '1'.
   - The interval values are loaded into the RM at this point.
   - The charger is started.
3. Delay for 100 µs.
4. Verify signal PQ is set to '1'.
   - The charger needs to set the output current to the PQ value and enable the charging process.
5. Battery model slowly increases voltage to 2.5 V, with an increase of 10 mV over a time of 200 µs.
6. The charger must stop the charging process, when measured voltage value is over set limit.

8.1.5 Timer Error

This test is designed to verify the ability to stop the charging process if the timer reaches 0.

Requirements covered in this scenario: Req.7.2.5.1,**Req.7.2.5.2.1**, **Req.7.2.5.2 Req.7.4.1**, **Req.7.4.1.1**,

Scenario description:

1. Set the battery voltage at 2.0 V, the temperature is set in the middle of the normal operation range (30 °C).
2. Verify signal Ready is set to '1'. The charger is started.
    - The interval values are loaded into the RM at this point. The timeout value for the PQ phase is set to 100 µs.
3. Delay for 100 µs.
4. Verify signal PQ is set to '1'.
    - The charger needs to set the output current to the PQ value and enable the charging process.
5. Battery model slowly increases voltage to 2.5 V, with an increase of 10 mV over a time of 200 µs.
6. Timeout should occur and charger should stop the charging process and not resume it.

8.2 PSL

Below are some properties and the requirements they cover. The default clock was set to the system clock for all PSL assertions and properties.

Property 8.2: Ensure that the timer value (charger and measurement) does not underflow. This property covers the requirements: **Req.7.4.1** and **Req.7.4.1.1**.

Listing 8.2: Underflow propery of timer.

```
-- psl property assert_timer_underflow is always ((
--      {
--        (cnt_q=C_zero_i) AND
--        ((cnt_en_i='1')  AND (cnt_ld='0'))
--      } |=> (cnt_q=C_zero_i)
--    ) async_abort rst_n='0'
-- );
```

Property 8.3: Checks that after starting, the SAR generates a ready signal after some time.

Listing 8.3: Propery checking if value was measured by SAR.

```
-- psl property assert_measurement is always (
--    (
--      {(start_en='1')} |=> {meas_ready_i[=1]}
--    ) abort (rst_n='0')
-- );
```

The property 8.4 is from "main" FSM, which controls the loading value to RM. It controls that the last address read from NVM memory was not higher than the last address in RM plus two for CRC bytes. The address generator in the NVM controller stops at one address higher than the last address read.

Listing 8.4: Underflow propery of timer.

```
-- psl property assert_last_address is never (
--    (address_q = (C_LAST_ADDR + 4))
-- );
```

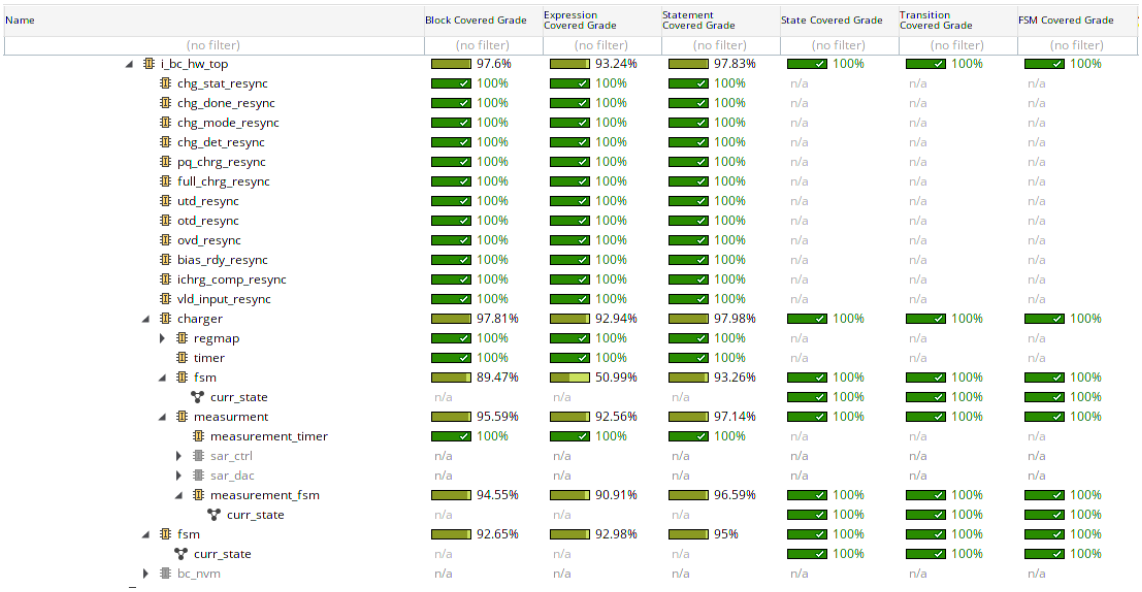
8.3 Coverage

The code coverage from all tests was merged and then analyzed in the program IMC (Integrated Metrics Center) (list. 8.5).

The tests were designed to toggle all lines at least once. This resulted in an overall coverage of 97.6 %, a detailed report can be seen in figure 8.1. The FSMs transitions, from cover data can be seen in figures 8.2 and 8.3.

Listing 8.5: Version of Integrated Metrics Center used.

```
>imc -version
IMC: 21.03-s003: (c) Copyright 1995-2021 Cadence Design
   Systems Inc ...
```



Name	Block Covered Grade	Expression Covered Grade	Statement Covered Grade	State Covered Grade	Transition Covered Grade	FSM Covered Grade
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
i_bc_hw_top	97.6%	93.24%	97.83%	100%	100%	100%
chg_stat_resync	100%	100%	100%	n/a	n/a	n/a
chg_done_resync	100%	100%	100%	n/a	n/a	n/a
chg_mode_resync	100%	100%	100%	n/a	n/a	n/a
chg_det_resync	100%	100%	100%	n/a	n/a	n/a
pq_chrg_resync	100%	100%	100%	n/a	n/a	n/a
full_chrg_resync	100%	100%	100%	n/a	n/a	n/a
utd_resync	100%	100%	100%	n/a	n/a	n/a
otd_resync	100%	100%	100%	n/a	n/a	n/a
ovd_resync	100%	100%	100%	n/a	n/a	n/a
bias_rdy_resync	100%	100%	100%	n/a	n/a	n/a
ichrg_comp_resync	100%	100%	100%	n/a	n/a	n/a
vld_input_resync	100%	100%	100%	n/a	n/a	n/a
charger	97.81%	92.94%	97.98%	100%	100%	100%
regmap	100%	100%	100%	n/a	n/a	n/a
timer	100%	100%	100%	n/a	n/a	n/a
fsm	89.47%	50.99%	93.26%	100%	100%	100%
curr_state	n/a	n/a	n/a	100%	100%	100%
measurment	95.59%	92.56%	97.14%	100%	100%	100%
measurement_timer	100%	100%	100%	n/a	n/a	n/a
sar_ctrl	n/a	n/a	n/a	n/a	n/a	n/a
sar_dac	n/a	n/a	n/a	n/a	n/a	n/a
measurement_fsm	94.55%	90.91%	96.59%	100%	100%	100%
curr_state	n/a	n/a	n/a	100%	100%	100%
fsm	92.65%	92.98%	95%	100%	100%	100%
curr_state	n/a	n/a	n/a	100%	100%	100%
bc_nvm	n/a	n/a	n/a	n/a	n/a	n/a

Fig. 8.1: Collected code coverage metrics and their grades for the hardware charger.

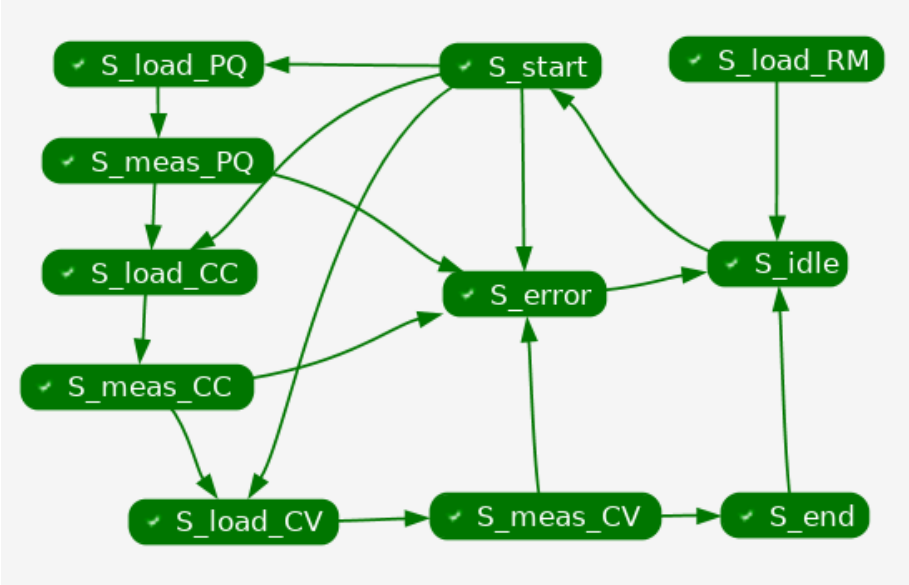


Fig. 8.2: Charger FSM states from coverage database

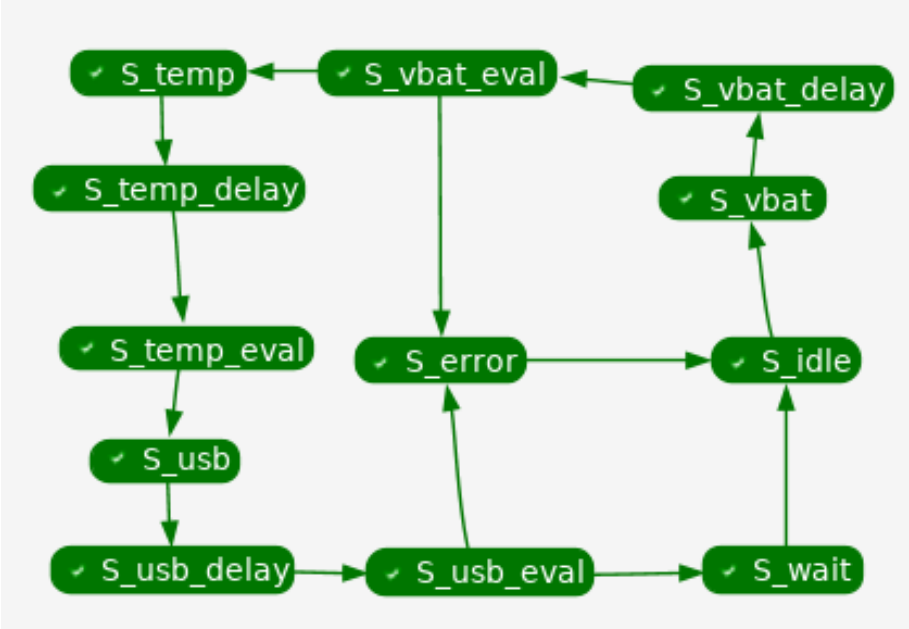


Fig. 8.3: Measurement FSM states from coverage database

8.4 Formal verification

Jasper (in version list. 8.6) is a tool supporting several types of automated formal verification for different applications (CDC, RDC, SuperLint, Control and Status Registers, ...) but only SuperLint was used. The CDC (and RDC) was not used because only one clock domain is used.

Listing 8.6: Version of Jasper Gold used.

```
>jg -version
2024.09p001 64 bits
```

8.4.1 SuperLint

SuperLint was run with almost all rules enabled; a complete list of all rules used in the linting design is in the Appendix B.

Jasper SuperLint outputted many warnings about deadlocks, livelocks, and state transitions as seen in table 8.1. However, all were reviewed and waived because they were designed to function this way.

In Fig. 8.4, Jasper Gold reported that the state *S_load_PQ* does not have any deadlock condition, meaning that the charger FSM cannot stay in this state and must jump to the next state on the next rising edge of the clock. This is the intended design behaviour of the charger, loading the value (for current) to the output in this state, and starting the measurement in the next state *S_meas_PQ*.

Table 8.1: Result of SuperLint.

Category	**Error**	**Warning**	**Info**	**Waived**	**Total**
FSM_DEADLOCK_LIVELOCK	0	0	0	22	22
FSM_REACHABILITY	0	0	0	9	9
Total	0	0	0	31	31

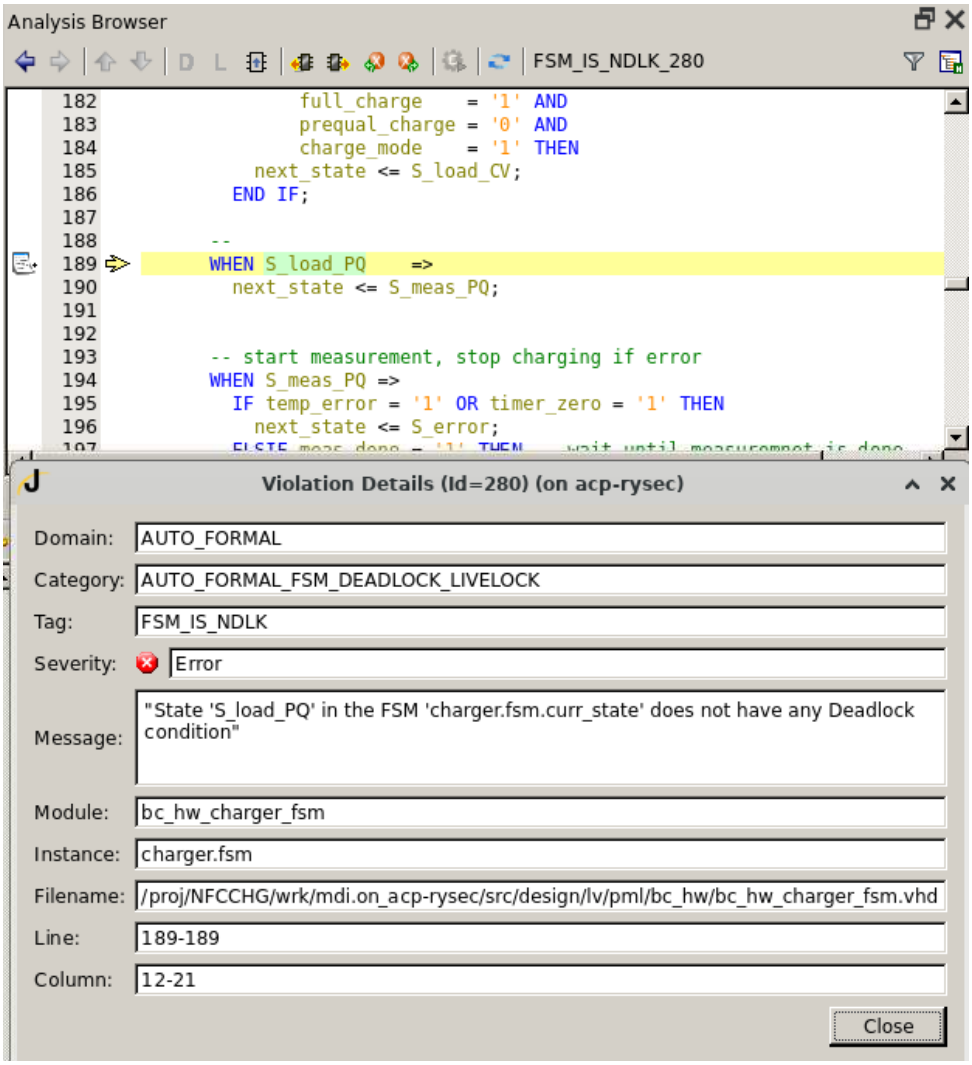


Fig. 8.4: Example of error reported by Jasper.

9 Synthesis

The synthesis was run to check approximate area utilization of the hardware charger. For synthesis, the Synopsis Design Compiler (lis. 9.1) was used. The extracted values from the synthesis area report can be seen in table 9.1. The entire (only about RTL described in this thesis) report about the area is in the appendix C.

In lis. 9.2 and 9.4 are states encoding for FSMs. The setting state of encoding for dc_shell was left at the default value, for both FSMs, dc_shell picked binary encoding.

Static timing analysis reports were generated for multiple scenarios, worst case and best case for clock 6 MHz + 15 % at duty cycle 0.5 ± 10 % and worst case and best case for clock 4 MHz + 10 %, duty cycle 0.5 ± 10 %). In lis. 9.3 is an example extracted from synthesis timing report for best case frequency at 6 MHz. It shows the path from one DFF to another DFF inside a timeout timer, showing that setup timing was met. Also, all the timing reports were reviewed and it was confirmed that no violation was reported for the post-synthesis gate-level netlist of the new "bc_hw block".

Synthesis automatically inserted clock gating for power consumption reduction, as shown in figure 9.1. Automatic clock insertion enabled for all blocks, except for register map.

Listing 9.1: Version of Design Compiler (dc_shell) used.

```
>dc_shell -version
dc_shell version    -  V-2023.12-SP5-1
dc_shell build date -  Sep 03, 2024
```

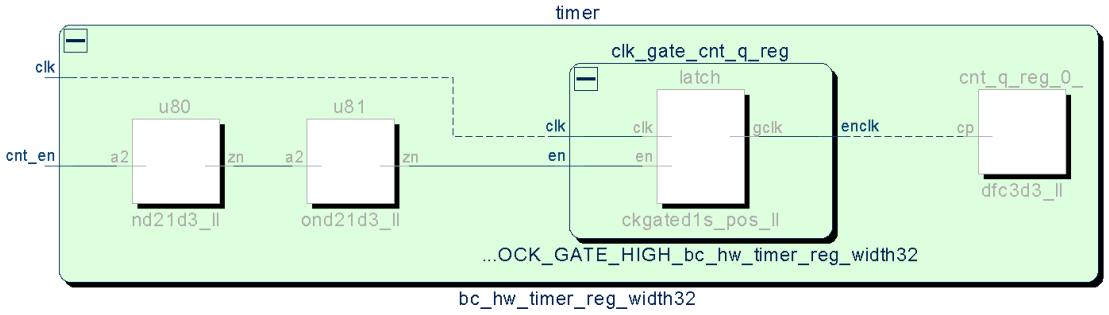


Fig. 9.1: Clock gating for timeout timer.

Table 9.1: Area size data from synthesis

Hierarchical	**Cell Area [um^2]**
bc_hw_top	55238.2
bc_hw_top/fsm	2939.7
bc_hw_top/nvm_ctrl	10490.4
bc_hw_top/charger	39344.1
bc_hw_top/charger/fsm	1653.1
bc_hw_top/charger/timer	3987.2
bc_hw_top/charger/regmap	20182.8
bc_hw_top/charger/measurement	10542.4
bc_hw_top/charger/measurement/fsm	3126.8
bc_hw_top/charger/measurement/SAR	2588.8

Listing 9.2: State coding for Charger FSM

```
statistics for FSM inference:
state register: curr_state
states
==========================
S_idle:                  0000
S_load_RM:               0001
S_start:                 0010
S_load_PQ:               0011
S_meas_PQ:               0100
S_load_CC:               0101
S_meas_CC:               0110
S_load_CV:               0111
S_meas_CV:               1000
S_end:                   1001
S_error:                 1010

total number of states: 11
```

Listing 9.3: Timing report from charger timer.

```
Startpoint: charger/timer/cnt_q_reg_0_ (clocked by
   CLK_OSC_RC_SYS)
Endpoint: charger/timer/cnt_q_reg_1_   (clocked by
   CLK_OSC_RC_SYS)
Scenario: SCENARIO_BC_ACTIVE6M
Path Group: CLK_OSC_RC_SYS
Path Type: max

Point                                 Incr  Path Voltage
-----------------------------------------------------------
clock CLK_OSC_RC_SYS (rise edge)      0.00  0.00
clock network delay (ideal)           0.00  0.00
charger/timer/cnt_q_reg_0_/cp         0.00  0.00 r 1.40
charger/timer/cnt_q_reg_0_/q          0.46  0.46 f 1.40
charger/timer/u3/zn                   0.29* 0.75 r 1.40
charger/timer/u86/zn                  0.16* 0.91 f 1.40
charger/timer/u87/zn                  0.15* 1.06 r 1.40
charger/timer/cnt_q_reg_1_/d          0.00* 1.06 r 1.40
data arrival time          1.06

clock CLK_OSC_RC_SYS (rise edge)      164.00      164.00
clock network delay (ideal)           0.00        164.00
clock uncertainty                     -0.20       163.80
timer/cnt_q_reg_1_/cp                 0.00        163.80 r
library setup time                    -0.41       163.39
data required time         163.39
-----------------------------------------------------------
data required time         163.39
data arrival time          -1.06
-----------------------------------------------------------
slack (MET)                162.32
```

Listing 9.4: State coding for Measurement FSM

```
statistics for FSM inference:
state register: curr_state
states
=========================
S_idle:                 0000
S_vbat:                 0001
S_vbat_delay:           0010
S_vbat_eval:            0011
S_temp:                 0100
S_temp_delay:           0101
S_temp_eval:            0110
S_usb:                  0111
S_usb_delay:            1000
S_usb_eval:             1001
S_wait:                 1010
S_error:                1011

total number of states: 12
```

10 Final IP block

The battery charger is implemented as an IP block, intended for integration into larger systems. All charging phases PQ, CC, and CV are managed by the main FSM. The battery voltage, USB voltage and temperature measurements are performed using a SAR ADC, while register map stores intervals and current status signals from analog. The charger supports JEITA guidelines, with parameters automatically modified based on temperature to improve battery safety and lifespan. Compliance with industry-standard verification flows has been proven through thorough verification. The synthesis was performed to get approximate area of the IP block.

Final hardware design and requirements were peer-review, by multiple people from ASICentrum. Created IP block will be integrated into multiple future projects.

Conclusion

Charging lithium polymer batteries is a complex process that requires measuring various parameters throughout the charging cycle. To extend battery life and prevent damage, it is essential to follow the manufacturer's recommendations outlined in the JEITA guidelines.

This work examines the relevant parameters for charging lithium polymer batteries and EDLCs. The discussion covers all stages of the charging process and details specific parameters, including maximum current values, voltage ranges, temperature limits, and time constraints. The hardware charger developed in this thesis is designed as an IP block, making it easy to integrate into any project requiring rechargeable power sources, such as lithium polymer batteries or EDLCs.

The design is divided into three main components. The main block (charger top-level) facilitates connections between the individual sub-blocks and the external environment, ensuring signal synchronization. One sub-block is the register map, which stores parameters for the charging process and incorporates four temperature zones in accordance with JEITA guidelines. The primary FSM in the top-level control manages the transition between charging phases, sets the output current, and periodically activates the measurement sub-block. This sub-block measures voltage (for the battery or EDLC), temperature, and the USB power source. The measurement block includes its own FSM for control and features a SAR, which is a reused IP from another project. The measurement FSM initiates the SAR, compares the measured values to the limits defined in the register map, and signals the main FSM if JEITA guidelines need to be activated, such as reducing the current or extending the charging time. The timer block is another crucial part of the design, functioning as a binary decrementing counter. It is utilized twice: once in the main block and once in the measurement block.

The developed design is integrated into a larger system that originally used software for charging. To facilitate this integration, a middle layer was created that includes an NVM controller for loading values into the register map. This controller is sourced from a previous project. Additionally, a small FSM was designed to store values from the NVM controller into the register map. The integrated design underwent verification through industry-standard methods. Functional tests, conducted under various scenarios and PSL properties, confirmed that the hardware battery charger functioned equivalently to the software implementation while adhering to JEITA guidelines. A formal linting structural analysis test demonstrated that the created RTL design complied with logical and structural rules. Furthermore, the synthesis report provided information about the design's size and confirmed that it met all timing constraints for hold and setup times.

Bibliography

[1] HORIBA, Tatsuo. Lithium-Ion Battery Systems. *Proceedings of the IEEE* [online]. 16 May 2014, 2014(102), 939 - 950 [accessed. 2024-10-25]. Available at: doi:10.1109/JPROC.2014.2319832

[2] Charging LiPo Batteries Safely: Principles and Practices. *Microbit Design* [online]. 2023 [accessed. 2025-04-01]. Available at: `https://www.microbitdesign.com/blog/lipo-charging`

[3] Lithium-ion Battery – en. Electricity - Magnetism [online]. 2024 [accessed. 2024-10-25]. Available at: https://www.electricity-magnetism.org/lithium-ion-battery-en/

[4] Li-ion battery materials: present and future. *Materials Today* [online]. 2015, 2015(5), 252 - 264 [accessed. 2024-10-30]. ISSN 1369-7021. Available at: doi:10.1016/j.mattod.2014.10.040

[5] Lessons Learned from the 787 Dreamliner Issue on Lithium-Ion Battery Reliability. Energies [online]. 2013, 2013(6), 4682 - 4695 [accessed. 2024-10-30]. Available at: doi:10.3390/en6094682

[6] Polymer Electrolytes for Lithium-Based Batteries: Advances and Prospects. *Chem* [online]. 2019, 2019(5), 2326 - 2352 [accessed. 2024-10-30]. ISSN 2451 - 9294. Available at: doi:10.1016/j.chempr.2019.05.00

[7] KAZDA, Tomáš. *Elektrochemické zdroje pro RAPS* [online]. 2024. Available at: `https://moodle.vut.cz/pluginfile.php/803494/mod_folder/content/0/Elektrochemick%C3%A9%20zdroje%20pro%20RAPS_dle%20%C5%A0ablony.pdf?forcedownload=1`

[8] NGUYEN-VAN, Hao, Dat NGUYEN, Thang NGUYEN, Minh NGUYEN a Loan PHAM-NGUYEN. *A Li-Ion battery charger with stable charging mode controller in noise environments* [online]. 25 January 2016. IEEE, 2016 [accessed 2025-10-25]. ISBN 978-1-4673-8374-5. Available at: doi:10.1109/ATC.2015.7388333

[9] JAPAN ELECTRONICS AND INFORMATION TECHNOLOGY INDUSTRIES ASSOCIATION. *A Guide to the Safe Use of Lithium-Ion Secondary Cells in Notebook-type Personal Computers and Tablet Terminals* [online]. 1 February 2021 [accessed. 2024-10-30]. Available at: `https://home.jeita.or.jp/upload_file/20210407112651_JbDgUSWGiB.pdf`

[10] NGUYEN-VAN, Hao, Dat NGUYEN, Thang NGUYEN, Minh NGUYEN a Loan PHAM-NGUYEN. *A Li-Ion battery charger with stable charging mode controller in noise environments* [online]. 25 January 2016. IEEE, 2016 [accessed. 2024-10-30]. ISBN 978-1-4673-8374-5. Available at: doi:10.1109/ATC.2015.7388333

[11] SVASTA, P., R. NEGROIU a Al. VASILE. Supercapacitors — *An alternative electrical energy storage device. In: 2017 5th International Symposium on Electrical and Electronics Engineering (ISEEE)* [online]. IEEE, 2017 [accessed. 2024-10-30]. ISBN 978-1-5386-2059-5. Available at: doi:10.1109/ISEEE.2017.8170626

[12] OKAZAKI, Yuhei, Masanobu YOSHIDA a Kenichiro FUJIWARA. Charging method of EDLCs by wind power generation in stand alone system. In: [online]. Sapporo, Japan, 09 August 2010 [accessed. 2024-10-25]. ISBN 978-1-4244-5393-1. Available at: doi:10.1109/IPEC.2010.5542358

[13] Supercapacitors Technical and Physical Basics of EDLC. In: *Würth Elektronik* [online]. 2019 [accessed. 2024-11-01]. Available at: https://www.psma.com/sites/default/files/uploads/files/Supercapacitors%20Technical%20and%20Physical%20Basics%20of%20EDLC%20(Rene%20Kalbitz%2C%20Wurth%20Elektronik).pdf

[14] HUN, Pui-yan, Huihui ZHANG, Han LIN a Qiaoshi GUO. Specializing liquid electrolytes and carbon-based materials in EDLCs for low-temperature applications. In: *Journal of Energy Chemistry* [online]. 2022, s. 580 - 602 [accessed. 2024-10-30]. Available at: doi:10.1016/j.jechem.2021.12.012

[15] WÜRTH ELEKTRONIK. *Supercapacitor – A Guide for the Design-In Process* [online]. [accessed. 2024-11-01]. Available at: https://www.we-online.com/components/media/o185469v410%20ANP077_Supercapacitor_A_Guide_for_the_Design-In_Process.pdf

[16] TEXAS INSTRUMENTS. *Li-ion battery-charger solutions for JEITA compliance* [online]. 2010 [accessed. 2024-10-29]. Available at: https://www.ti.com/lit/an/slyt365/slyt365.pdf?ts=1730362828776

[17] MASSACHUSETTS INSTITUTE OF TECHNOLOGY. *Finite State Machine* [online]. 2017 [accessed. 2024-11-04]. Available at: https://web.mit.edu/6.111/www/f2017/handouts/L06.pdf

[18] PONG P., Chu. *RTL Hardware Design Using VHDL: Coding for Efficiency, Portability, and Scalability* [online]. IEEE press, 2006 [accessed. 2025-05-17]. ISBN 9780471786399. Available at: doi:10.1002/0471786411

[19] MIELCAREK, Kamil, Alexander BARKALOV a Larysa TITARENKO. Designing Moore FSM with extended class codes. In: *2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)* [online]. Thessaloniki, Greece: IEEE, 2017, s. 1 - 4 [accessed. 2025-05-17]. Available at: doi:10.1109/MOCAST.2017.7937623

[20] MUSIL, Vladislav a Jaromír KOLOUCH. *Návrh digitálních integrovaných obvodů VLSI a jazyk VHDL* [online]. 2007 [accessed. 2024-10-30]. Available at: `https://moodle.vut.cz/pluginfile.php/842149/mod_resource/content/3/Navrh_digitalnich_integrovanych_obvodu_a_jazyk_VHDL_S.pdf`

[21] CUMMINGS, CLIFFORD E. *State Machine Coding Styles for Synthesis* [online]. 1998 [accessed. 2025-05-17]. Available at: `http://www.sunburst-design.com/papers/CummingsSNUG1998SJ_FSM.pdf`

[22] XILINX. *Modeling Registers and Counters* [online]. 2015 [accessed. 2024-11-01]. Available at: `https://www.amd.com/content/dam/amd/en/documents/university/vivado-teaching/hdl-design/2015x/VHDL/docs-pdf/lab6.pdf`

[23] ŠŤASTNÝ, Jakub. *FPGA prakticky : realizace číslicových systémů pro programovatelná hradlová pole.* Praha: BEN - technická literatura, 2011. ISBN 978-80-7300-261-9.

[24] IEEE Standard for Property Specification Language (PSL). In: IEEE Std 1850-2005 [online]. 2005, s. 1-143 [accessed. 2025-04-25]. Available at: doi:10.1109/IEEESTD.2005.97780

[25] HANY, Amr, Ahmed ISMAIL, Ahmed KAMAL a Mohamed BADRAN. Approach for a unified functional verification flow. In: *2013 Saudi International Electronics, Communications and Photonics Conference* [online]. Riyadh, Saudi Arabia: IEEE, 2013 [accessed. 2025-05-17]. Available at: doi:10.1109/SIECPC.2013.6550753

[26] GIRISH, M, G GOPAKUMAR a D S DIVYA. Formal and Simulation Verification: Comparing and Contrasting the two Verification Approaches. In: *2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)* [online]. Ernakulam, India: IEEE, 2021 [accessed. 2025-05-17]. ISBN 978-1-7281-7136-4. Available at: doi: https://doi.org/10.1109/ACCESS51619.2021.9563305

[27] CADENCE. *Jasper CDC App* [online]. 2025 [accessed. 2025-05-02]. Available at: `https://www.cadence.com/en_US/home/tools/system-design-and-verification/formal-and-static-verification/jasper-verification-platform/jaspergold-clock-domain-crossing-app.html`

[28] CADENCE. *Jasper CDC App* [online]. 2025 [accessed. 2025-05-02]. Available at: `https://www.cadence.com/en_US/home/tools/system-design-and-verification/formal-and-static-verification/jasper-verification-platform/jaspergold-clock-domain-crossing-app.html`

[29] SHCHETINIUK, Kirill. *Analogue Circuit Modeling and Verification in SystemVerilog.* Online, bachelor's Thesis. Marcela ZACHARIÁŠOVÁ (supervisor). Brno: Brno University of Technology, Faculty of Information Technology, 2025. Available at: https://www.vut.cz/en/students/final-thesis/detail/164589. [accessed 2025-05-20].

Symbols and abbreviations

Li-ion	Lithium ion
Li-pol	Lithium polymer
EDLC	Electric double layer capacitor
CPU	Central processing unit
SAR	Successive-approximation-register
DAC	Digital to analog converter
NTC	Negative temperature coefficient
PTC	Positive temperature coefficient
RM	Register Map
DUT	Device under test
PSL	Property Specification Language
CDC	Clock Domain Crossing
RTL	Register Transfer Level
RDC	Reset Domain Crossing
VLSI	Very Large Scale Integration
SVA	System Verilog Assertions
SoC	System on Chip
NVM	Non-Volatile Memory
LSB	Least Significant Bit
MSB	Most Significant Bit
IP	Intellectual Property
SV-RNM	SystemVerilog - Real Number Modeling

List of appendices

A Content of the electronic appendix **69**

B SuperLint setting **70**

C Synthesis Data **72**

A Content of the electronic appendix

```
bc_hw
├── bc_hw_charger_fsm.vhd
├── bc_hw_charger_top.vhd
├── bc_hw_fsm.vhd
├── bc_hw_measurement_fsm.vhd
├── bc_hw_measurement_top.vhd
├── bc_hw_reg_map.vhd
├── bc_hw_timer.vhd
└── bc_hw_top.vhd
```

B SuperLint setting

Settings for SuperLint that were used to verified design.

FIL_MS_DUNM	CST_NO_IVAL	LOP_NR_MLPV	RST_IS_CPLX
FIL_NR_MMOD	CST_NR_REAL	LOP_NR_MSTP	SIG_NO_ASIG
FIL_NS_SUFX	DLY_NO_CSAG	MAC_NR_DMUL	TSK_NR_ESDE
ALW_NR_EMSL	DLY_NR_NEGT	MOD_NO_LGIC	VAR_NO_ASIG
ASG_NR_FNCP	DUT_NO_USED	MOD_NR_EBLK	VAR_NR_NEGT
ASG_NR_EXPR	EXP_NR_DIVZ	MOD_NR_INCD	VAR_NR_OUTL
ASG_NR_EVNT	EXP_NR_ITYC	MOD_NR_PGAT	VAR_NR_PRCD
MOD_NR_STRN	EXP_NR_OVFB	MOD_NR_PRIM	VAR_NR_RDBA
ARY_MS_DRNG	EXP_NR_MXSU	MOD_NR_RPXZ	VAR_NR_RDEC
ARY_NR_LOPR	EXP_NR_STUS	MOD_NR_UNGN	VAR_NR_STRN
ARY_NR_NERG	EXP_NR_TNST	REP_NR_ZERO	VAR_NR_UBDL
ARY_NR_SLRG	EXP_NR_USTS	NAM_NR_REPU	WIR_NR_IMPL
ASG_MS_RTRU	FNC_MS_AFPR	OPR_NR_LOSD	WIR_NR_UASR
ASG_MS_VALR	FNC_MS_MTYP	OPR_NR_REAL	ALW_NR_MSLV
ASG_NR_POVF	FNC_NO_AVAC	OPR_NR_STRE	ALW_NR_UNUV
ASG_NR_LMSB	FNC_NO_UASG	OPR_NR_UCMP	CAS_NR_FPIN
ASG_NR_MINP	FNC_NR_AVGV	OPR_NR_UEOP	CAS_NR_PUOV
ASG_NR_NBFC	FNC_NR_NARG	OPR_NR_UEAS	CST_NR_MSBX
ASG_NR_SOVF	FNC_NR_RLRT	OPR_NR_TERN	CST_NR_MSBZ
CAS_NO_CNST	FNC_NR_UGLV	OPR_NR_UREL	DLY_NR_XZVL
CAS_NO_DEFA	FNC_NR_SYSV	OTP_NO_FDRV	ENM_NR_TOST
CAS_NO_OTHR	GEN_NO_GREP	OTP_NR_READ	FNC_NO_EVAL
CAS_NR_CMUL	IFC_NO_FALW	OTP_NR_TSUP	MOD_NR_ACCD
CAS_NR_DEFN	IFC_NR_DGEL	OTP_NR_UDRV	MOD_NR_LOCL
CAS_NR_EXCS	INP_NO_LOAD	OTP_UC_INST	MOD_NR_SYXZ
CAS_NR_OVCI	INP_UC_INST	PAR_MS_SDAS	VAR_NR_MBNB
CAS_NR_UCIT	INP_NR_IOUT	PRG_NR_PROF	VAR_NR_NDCL
CAS_NR_XCAZ	INP_NR_TSUP	PRO_NR_WAIT	ALW_IC_SENL
CND_NR_BWOP	INS_MS_PRMS	PRT_NO_MODE	ALW_NO_COMB
CND_NR_DUPF	INS_MS_PSIZ	PRT_NR_BUFF	ALW_NO_ETRG
CND_NR_EVXZ	INS_NR_PODL	PRT_NR_DFRG	ALW_NO_EVTS
CND_NS_MBEX	INS_NR_PTEX	PRT_NR_LINK	ALW_NR_MCLK
CST_MS_SIZE	IOP_NR_UASG	REG_NR_UASR	ALW_NR_MIFF
CST_NO_DELY	LOP_NR_CTCE	REP_NR_NPOS	ALW_NR_MXCK
CST_NO_HIER	LOP_NR_FCND	REP_NR_UPAC	ALW_NR_TCST

ASG_NR_NBCB
ASG_NR_SUPN
BLK_NR_PGDL
CAS_NR_EVLX
CLK_IS_NSYT
CLK_NR_DDBE
CLK_NR_EDGE
CND_NR_CMXZ
CST_NR_CCAT
NOW_NS_FUNC
FLP_NR_ASMX
FLP_NR_BLAS
FLP_NR_MBCK
FNC_NR_CREC
FNC_NR_RESL
IDX_NR_LBOU
IDX_NR_ORNG
IFC_NO_SENL
INP_NR_ASGN
INS_NR_TOKN
LAT_NR_BLAS
LAT_NR_MXCB
LOP_NR_GLID
LOP_NR_INFL
LOP_NR_SRLG
MOD_IS_RCMP
MOD_NR_ALAS
MOD_NR_ASLD
MOD_NR_CNDO
MOD_NR_DATT
MOD_NR_DSBC
MOD_NR_EVRP
MOD_NR_FKJN
MOD_NR_SPFY
MOD_NR_FORE
MOD_NR_FREL
MOD_NR_IFSM
MOD_NR_NSLP
MOD_NR_NULP
MOD_NR_PHYD
MOD_NR_USWC
MOD_NS_ADAS
MOD_NS_DCSP
MOD_NS_GTIN
OPR_NR_WCCO
SIG_NO_HIER
SIG_NO_UHIR
SIG_NR_MDRV
SIG_NR_TIME
TSK_NR_CLKE
VAR_NO_COMR
VAR_NR_MBLA
VAR_NR_MNBA
VAR_NR_OUTR
VAR_NR_REAL
CLK_NO_INPT
CLK_NS_EDMX
CLK_NR_GLKN
CMB_NR_TLIO
FLP_NO_ASRT
FLP_NR_ASRT
FLP_NR_ENCT
FLP_IS_STRH
NET_NO_DRIV
RST_IS_DCMB
RST_IS_DFLP
RST_IS_DLAT
RST_NR_PENA
SIG_IS_MDRV
ELB_IS_ERRO
INS_IS_EBXE
MOD_IS_EBXE
MOD_IS_IBXE
MOD_IS_RBXE
SIG_IS_ATBX
FLP_IS_CDFF
FLP_IS_GTCK
FLP_NO_SRST
LAT_IS_INFR
OTP_NO_RGTM
RST_IS_DDAF
FSM_IS_LVLK
FSM_IS_DDLK
FSM_IS_PLLK
FSM_IS_PDLK
FSM_IS_NLLK
FSM_IS_NDLK
FSM_NO_RCHB
FSM_NO_TRRN

C Synthesis Data

Cell area data (in μm^2), only blocks created in this thesis or used as an IPs.

```
i_bc_hw_top 55238.1990
bc_nvm 10490.3713
bc_nvm/crc_ena_resync 67.5792
bc_nvm/i_bc_nvm_crc 1809.0433
bc_nvm/i_bc_nvm_crc/clk_gate_crc_reg 57.1824
bc_nvm/i_bc_nvm_crc/i_bc_ck_gate_or 23.3928
bc_nvm/i_bc_nvm_crc/i_bc_ck_gate_or/i_or2 15.5952
bc_nvm/i_cnt_sync 98.7696
bc_nvm/i_gmux2_nvm_bg_int_en 23.3928
bc_nvm/i_gmux2_nvm_clk_out_en 23.3928
bc_nvm/i_gmux2_nvm_erase 23.3928
bc_nvm/i_gmux2_nvm_hv_enable 23.3928
bc_nvm/i_gmux2_nvm_iref_en 23.3928
bc_nvm/i_gmux2_nvm_osc_en 23.3928
bc_nvm/i_gmux2_nvm_read 23.3928
bc_nvm/i_gmux2_nvm_write 23.3928
bc_nvm/i_nvm_agen 1083.8664
bc_nvm/i_nvm_agen/clk_gate_nvm_wadd_q_reg 57.1824
bc_nvm/i_nvm_ctrl_fsm 3607.6896
bc_nvm/i_nvm_ctrl_fsm/clk_gate_m_curr_state_reg 57.1824
bc_nvm/i_nvm_data_reg 610.8120
bc_nvm/i_nvm_data_reg/clk_gate_nvm_updt_data_q_reg 57.1824
bc_nvm/i_nvm_req_sel 309.3048
bc_nvm/i_nvm_slave_fsm 142.9560
bc_nvm/i_nvm_timer 1715.4720
bc_nvm/i_nvm_timer/clk_gate_nvm_timer_q_reg 57.1824
bc_nvm/i_rm_nvm_addr_wr_gmux2 23.3928
bc_nvm/nvm_enable_clk_gmux2 23.3928
bc_nvm/rm_nvm_data_wr_gmux2 23.3928
bc_nvm/updt_req_resync 184.5432
bc_nvm/updt_req_resync/i_ck_gate 15.5952
bc_nvm/updt_req_resync/i_ck_gate/i_or2 15.5952
bc_nvm/updt_req_resync/i_ena_gate_for2 0.0000
bias_rdy_resync 135.1584
charger 39344.0908
charger/fsm 1653.0912
```

```
charger/fsm/clk_gate_curr_state_reg 57.1824
charger/measurment 10542.3552
charger/measurment/measurement_fsm 3126.8376
charger/measurment/measurement_fsm/clk_gate_curr_state_reg 57.1824
charger/measurment/measurement_fsm/clk_gate_measur_temp_q_reg
57.1824
charger/measurment/measurement_timer 2604.3984
charger/measurment/measurement_timer/clk_gate_cnt_q_reg 57.1824
charger/measurment/sar_ctrl 1733.6664
charger/measurment/sar_ctrl/clk_gate_sys_timer_val_reg 57.1824
charger/measurment/sar_ctrl/clk_gate_sys_vld_int_reg 57.1824
charger/measurment/sar_dac 2588.8032
charger/measurment/sar_dac/clk_gate_vld_level_i_reg 57.1824
charger/measurment/sar_dac/clk_gate_vld_level_or_reg 57.1824
charger/measurment/sar_dac/i_gmux2 23.3928
charger/regmap 20182.7883
regmap/i_addr_dec_dp_bnk0 1393.1712
regmap/i_reg_rd_bc_hw 3220.4089
regmap/i_regt_rw_bat_otd_en 93.5712
regmap/i_regt_rw_bat_otd_en/i_ck_gate 12.9960
regmap/i_regt_rw_bat_ovd_en 93.5712
regmap/i_regt_rw_bat_ovd_en/i_ck_gate 12.9960
regmap/i_regt_rw_bat_utd_en 93.5712
regmap/i_regt_rw_bat_utd_en/i_ck_gate 12.9960
regmap/i_regt_rw_chg_block_en 93.5712
regmap/i_regt_rw_chg_block_en/i_ck_gate 12.9960
regmap/i_regt_rw_chg_level 288.5112
regmap/i_regt_rw_chg_level/i_ck_gate 12.9960
regmap/i_regt_rw_chg_top_off_lim 161.1504
regmap/i_regt_rw_chg_top_off_lim/i_ck_gate 12.9960
regmap/i_regt_rw_chg_trim_ib 431.4672
regmap/i_regt_rw_chg_trim_ib/i_ck_gate 12.9960
regmap/i_regt_rw_chg_trim_rsi 499.0464
regmap/i_regt_rw_chg_trim_rsi/i_ck_gate 12.9960
regmap/i_regt_rw_htd_en 93.5712
regmap/i_regt_rw_htd_en/i_ck_gate 12.9960
regmap/i_regt_rw_jeita_en 93.5712
regmap/i_regt_rw_jeita_en/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_cc_dur_lsb 566.6256
```

regmap/i_regt_rw_reg_hw_cc_dur_lsb/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_cc_dur_msb 566.6256
regmap/i_regt_rw_reg_hw_cc_dur_msb/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_common 566.6256
regmap/i_regt_rw_reg_hw_common/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_crc_hi 566.6256
regmap/i_regt_rw_reg_hw_crc_hi/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_crc_lo 566.6256
regmap/i_regt_rw_reg_hw_crc_lo/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_cv_dur_lsb 566.6256
regmap/i_regt_rw_reg_hw_cv_dur_lsb/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_cv_dur_msb 566.6256
regmap/i_regt_rw_reg_hw_cv_dur_msb/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_full1 566.6256
regmap/i_regt_rw_reg_hw_full1/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_full2 566.6256
regmap/i_regt_rw_reg_hw_full2/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_full3 566.6256
regmap/i_regt_rw_reg_hw_full3/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_htd_trim 499.0464
regmap/i_regt_rw_reg_hw_htd_trim/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_pq_dur_lsb 566.6256
regmap/i_regt_rw_reg_hw_pq_dur_lsb/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_pq_dur_msb 566.6256
regmap/i_regt_rw_reg_hw_pq_dur_msb/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_t1 431.4672
regmap/i_regt_rw_reg_hw_t1/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_t2 431.4672
regmap/i_regt_rw_reg_hw_t2/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_t3 431.4672
regmap/i_regt_rw_reg_hw_t3/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_t4 431.4672
regmap/i_regt_rw_reg_hw_t4/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_test0 566.6256
regmap/i_regt_rw_reg_hw_test0/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_v_max_cc 431.4672
regmap/i_regt_rw_reg_hw_v_max_cc/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_v_max_cv 431.4672
regmap/i_regt_rw_reg_hw_v_max_cv/i_ck_gate 12.9960

```
regmap/i_regt_rw_reg_hw_v_max_pq 431.4672
regmap/i_regt_rw_reg_hw_v_max_pq/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_v_min 431.4672
regmap/i_regt_rw_reg_hw_v_min/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_v_trim1 363.8880
regmap/i_regt_rw_reg_hw_v_trim1/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_v_trim1_j 363.8880
regmap/i_regt_rw_reg_hw_v_trim1_j/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_v_trim2 363.8880
regmap/i_regt_rw_reg_hw_v_trim2/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_v_trim2_j 363.8880
regmap/i_regt_rw_reg_hw_v_trim2_j/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_v_trim3 363.8880
regmap/i_regt_rw_reg_hw_v_trim3/i_ck_gate 12.9960
regmap/i_regt_rw_reg_hw_v_trim3_j 363.8880
regmap/i_regt_rw_reg_hw_v_trim3_j/i_ck_gate 12.9960
timer 3987.1728
timer/clk_gate_cnt_q_reg 57.1824
chg_det_resync 135.1584
chg_done_resync 135.1584
chg_mode_resync 135.1584
chg_stat_resync 135.1584
fsm 2939.6952
fsm/clk_gate_address_q_reg 57.1824
fsm/clk_gate_curr_state_reg 57.1824
full_chrg_resync 135.1584
ichrg_comp_resync 135.1584
otd_resync 135.1584
ovd_resync 135.1584
pq_chrg_resync 135.1584
utd_resync 135.1584
vld_input_resync 135.1584
```