



**Politecnico  
di Torino**

**Corso di Laurea Magistrale  
in Ingegneria Elettronica**

**Tesi di Laurea Magistrale**

**Analisi, caratterizzazione e sviluppo di un controllore digitale USB  
Type-C per microcontrollori STM32**

**Relatori**

Prof. Danilo Demarchi  
Ing. Giuseppe Guarnaccia  
Ing. Giovanni Pangallo

**Candidato**

Giuseppe Bellarmino

Ottobre 2024

*A te Nonno,  
che più di tutti avresti voluto  
essere qui oggi.  
Ti porterò nel mio cuore,  
sempre.*

# Sommario

L'obiettivo di questa tesi è la progettazione di un controllore digitale integrato che implementa le funzionalità del protocollo **USB Type-C**.

La soluzione proposta consente l'integrazione delle funzionalità **Type-C** all'interno di un microcontrollore **STM32**, ottimizzando l'area.

Nella parte iniziale vengono illustrati i protocolli **USB** e la loro evoluzione nel tempo, per poi proseguire con l'analisi delle specifiche del protocollo **USB Type-C**.

Successivamente, si analizzano le soluzioni attuali di **USB Type-C** fornite nei microcontrollori **STM32** da **STMicroelectronics**.

La tesi prosegue con la progettazione del controllore digitale **Type-C Only**.

Infine, il controllore viene testato mediante simulazioni.

# Indice

<b>Elenco delle tabelle</b>	5
<b>Elenco delle figure</b>	6
<b>1 Introduzione ai protocolli Universal Serial Bus</b>	8
1.1 Terminologia USB	8
1.2 Protocolli dati USB	9
1.2.1 USB 1	9
1.2.2 USB 2	10
1.2.3 USB 3	11
1.2.4 USB 4	12
1.2.5 Confronto velocità e connettori USB	12
1.3 USB Power Delivery	14
<b>2 USB Type-C</b>	15
2.1 USB Type-C Plug e Receptacle	16
2.1.1 Descrizione Pin	16
2.2 Processo di configurazione	18
2.2.1 Connessioni funzionali tra Sources e Sinks	18
2.2.2 Configurazione della potenza	20
<b>3 USB Type C e Power Delivery nei Microcontrollori STM32</b>	21
3.1 Type-C e Power Delivery controller	21
3.1.1 Diagramma a Blocchi	21
3.2 Type-C senza UCPD	23
3.2.1 Type-C in modalità Device	23
3.2.2 Type-C in modalità Host	24
3.2.3 Type C in modalità OTG	25
<b>4 Sviluppo controllore Type-C only</b>	27
4.1 Input e Output	28
4.1.1 Input e Output del PHY	28
4.2 Clock e Reset	29
4.3 APB_ITF	30
4.3.1 Lettura e scrittura APB	30
4.3.2 Mappa dei registri	31
4.4 Clock_Divider	37
4.5 Sincronizzatore	38
4.6 Kernel	39
4.6.1 CC1 Finite State Machine	41
4.7 Wake-Up	42

<b>5</b>	<b>Simulazioni e Sintesi</b>	<b>43</b>
5.1	Simulazioni . . . . .	43
5.1.1	Test dei Registri . . . . .	43
5.1.2	Test Source . . . . .	48
5.1.3	Test Sink . . . . .	53
5.2	Sintesi . . . . .	56
5.2.1	Report Area . . . . .	56
5.2.2	Report Timing . . . . .	57
5.3	Conclusioni . . . . .	58

# Elenco delle tabelle

1.1	Pinout USB 1.1/2.0 . . . . .	9
1.2	Pinout connettore 3.0 Type-A, Type-B . . . . .	11
1.3	Velocità massima supportata dai protocolli USB . . . . .	12
1.4	Connessioni supportate USB [2] . . . . .	13
2.1	Pin di connessione per USB Type-C . . . . .	16
2.2	Interconnessioni USB Type-C . . . . .	18
2.3	Stato connessione dal punto di vista del Source . . . . .	19
2.4	Comportamento del Source e del Sink in base allo stato . . . . .	19
2.5	Ordine di precedenza delle modalità di erogazione della potenza . . . . .	20
4.1	Input dell'IP provenienti dal PHY CC1 . . . . .	28
4.2	Output dell'IP per il PHY CC1 . . . . .	29
4.3	Interfaccia APB . . . . .	30
4.4	VSTATE_CC1[1:0] . . . . .	39
4.5	Generazione segnali CC1_Level_xx . . . . .	40
4.6	Condizioni di reset del contatore CC1_Counter, della macchina a stati 1_FSM e dei segnali CC1LvlxxSeen . . . . .	41

# Elenco delle figure

1.1	USB 1.0/2.0 Plug [2]	9
1.2	USB 1.0/2.0 Receptacle [2]	9
1.3	USB Mini 2.0 Plug [2]	10
1.4	USB Mini 2.0 Receptacle [2]	10
1.5	USB Micro 2.0 Plug [2]	10
1.6	USB Micro 2.0 Receptacle [2]	10
1.7	USB 3.0 Plug [2]	11
1.8	USB 3.0 Receptacle [2]	11
1.9	Evoluzione Power Delivery	14
2.1	USB Type-C Full-Feature	16
2.2	Connessione Source-Sink [4]	18
3.1	USB Type-C Power Delivery (UCPD) [5]	22
3.2	STM32 come Device [4]	23
3.3	STM32 come Host [4]	24
3.4	STM32 come Legacy OTG [4]	26
4.1	Esempio lettura e scrittura del campo UCEN	31
4.2	CFG1	31
4.3	CFG2	32
4.4	CFG3	33
4.5	CR	34
4.6	IMR	35
4.7	SR	36
4.8	ICR	37
4.9	Transizione stati FSM	42
5.1	Reset dei registri	44
5.2	Test scrittura/lettura CFG1	44
5.3	Test scrittura/lettura CFG2	45
5.4	Test scrittura/lettura CFG3	45
5.5	Scrittura CR con IP disabilitata	46
5.6	Scrittura CR con IP abilitata	47
5.7	Configurazione PSC, HBITCLKDIV e abilitazione IP (UCEN)	48
5.8	Scrittura CR: ANAMODE = '0', ANASUBMODE = "01" e CCEnable = "01"	48
5.9	Input e Output PHY dopo la scrittura del CR	49
5.10	FSM STABLE_01	50
5.11	Scrittura TYPECEVT1 e TypeCVStateCC1 in SR	50
5.12	Passaggio ANASUBMODE da "01" a "10"	51
5.13	PHY ANASUBMODE = "10"	51
5.14	FSM dopo il cambiamento di ANASUBMODE	52
5.15	Scrittura all'interno del CR per configurare l'IP come Sink e abilitare la linea CC2	53

5.16	Input e Output PHY dopo aver impostato l'IP come Sink ed aver abilitato la linea CC2 . . . . .	54
5.17	FSM con CC2 a 1.5V . . . . .	54
5.18	Scrittura TYPECEVT2 e TYPECVSTATECC2 in SR, CC2_Voltage = 1.5V	55
5.19	Report area . . . . .	56
5.20	Report Critical Path PClk . . . . .	57
5.21	Report Critical Path USBClk . . . . .	58

# Capitolo 1

## Introduzione ai protocolli Universal Serial Bus

Nel 1994 un gruppo di sette società costituito da Compaq, DEC, Hewlett-Packard, IBM, Microsoft, NEC e Nortel decise di sviluppare uno standard industriale di comunicazione seriale. Venne così istituita l'organizzazione Universal Serial Bus Implementers Forum (USB-IF), nata con lo scopo di promuovere lo sviluppo della tecnologia USB. [1]

L'obiettivo è semplificare il collegamento tra i dispositivi esterni e i computer. Prima dell'introduzione del USB, venivano utilizzate porte parallele e seriali individuali per connettere periferiche come tastiere, mouse, controller e stampanti.

Lo standard USB definisce le specifiche per cavi, connettori e protocolli per la connessione, la comunicazione e l'alimentazione tra dispositivi e le loro periferiche.

### 1.1 Terminologia USB

Il termine **Host** indica il dispositivo principale che controlla la comunicazione USB, mentre il **Device** rappresenta qualsiasi periferica collegata all'**Host**. Quando si parla di alimentazione, il dispositivo che fornisce energia attraverso il connettore USB è chiamato **Source**. Tipicamente, l'**Host** funge anche da **Source**, ma con l'introduzione dello standard USB **Type-C**, anche i **Device** possono assumere questo ruolo. Il dispositivo che riceve l'energia è invece denominato **Sink**, tipicamente un **Device**.

Per quanto riguarda le porte USB, il termine **UFP** (**Upstream Facing Port**) indica una porta che si connette all'**Host**. Mentre, la **DFP** (**Downstream Facing Port**) è la porta che si collega a un **UFP** e gestisce la comunicazione verso le periferiche. Inoltre, le porte **DRP** (**Dual Role Port**) e **DRD** (**Dual Role Data**) possono assumere rispettivamente il ruolo di **Source** o **Sink** e di **Host** o **Device**, a seconda delle necessità.

## 1.2 Protocolli dati USB

### 1.2.1 USB 1

USB 1.0 venne lanciato nel gennaio del 1996, inizialmente destinato esclusivamente ai computer con Windows 95. La diffusione effettiva del protocollo avvenne con il rilascio di USB 1.1 nel 1998. Questo aggiornamento migliorò l'affidabilità delle connessioni, riducendo gli errori di trasmissione. USB 1.1 consentiva il trasferimento dei dati in **Low Speed (LS)**, a 1.5 Megabit per secondo (Mbps), e in **Full Speed (FS)**, a 12 Mbps. La modalità LS veniva utilizzata principalmente per dispositivi come tastiere e mouse, mentre la modalità FS era impiegata per floppy disk e periferiche come stampanti.

Erano presenti due tipi di connettori standard: **Type-A** e **Type-B**. All'interno del cavo e del connettore USB si trovano quattro linee/pin, come mostrato nelle figure 1.1 e 1.2, utilizzati sia per la comunicazione dati che per l'alimentazione dei dispositivi.

Come indicato nella tabella 1.1 la comunicazione, di tipo differenziale, utilizza i pin 2 e 3, mentre i pin 1 e 4 sono dedicati all'alimentazione.

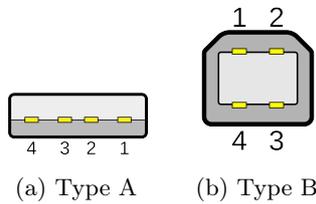


Figura 1.1: USB 1.0/2.0 Plug [2]

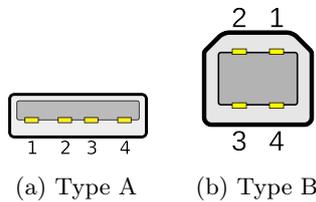


Figura 1.2: USB 1.0/2.0 Receptacle [2]

Pin	Nome segnale	Descrizione
1	VBUS	+5V
2	D-	Data-
3	D+	Data+
4	GND	Ground

Tabella 1.1: Pinout USB 1.1/2.0

### 1.2.2 USB 2

Dopo due anni, avviene il rilascio di USB 2.0, che introduce la velocità di comunicazione **High Speed (HS)**, con una velocità massima di 480 Mbps. Nel corso degli anni, la specifica ha ricevuto diversi aggiornamenti tramite **Engineering Change Notice (ECN)**.

**USB On-The-Go (USB OTG)** nasce come estensione del protocollo. Quest'ultimo consente alla periferica USB di funzionare sia in modalità **Host** sia in modalità **Device**. Ad esempio, uno smartphone con USB OTG può operare come **Host** per leggere o scrivere da una chiavetta USB, utilizzare mouse, tastiere o controller, ma può anche funzionare come **Device** quando viene connesso a un computer.

Nel protocollo USB 2.0 continuano a essere utilizzati i connettori **Type-A** e **Type-B** (fig. 1.1 e 1.2) del precedente protocollo e vengono introdotti i connettori **Mini** (fig. 1.3 e 1.4) e **Micro** (fig. 1.5 e 1.6). Questi ultimi presentano un pin in più, utilizzato per l'identificazione del ruolo dati dalla USB OTG.

L'alimentazione fornita dalla USB 2.0 in modalità **Host** è di 5V e garantisce una corrente massima di 500 mA.



Figura 1.3: USB Mini 2.0 Plug [2]

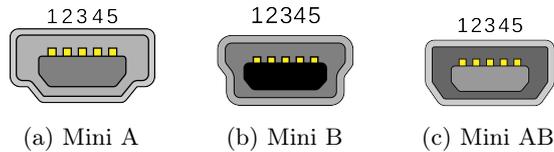


Figura 1.4: USB Mini 2.0 Receptacle [2]

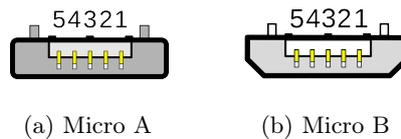


Figura 1.5: USB Micro 2.0 Plug [2]

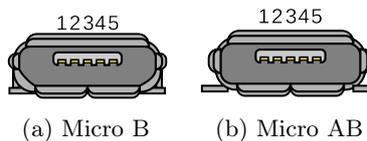


Figura 1.6: USB Micro 2.0 Receptacle [2]

### 1.2.3 USB 3

Nel 2008 viene rilasciato il protocollo USB 3.0, che introduce la modalità **SuperSpeed** (SS) con velocità di trasferimento fino a 5 Gbps supportando la modalità full-duplex. Questo permette la trasmissione e la ricezione dei dati simultaneamente.

Vengono introdotti nuovi connettori **Type-A**, **Type-B** e **Micro-B** (fig. 1.7 e 1.8), riconoscibili dal colore blu della plastica interna. Il nuovo connettore **Type-A** mantiene la compatibilità con il connettore **Type-A** di USB 2.0, permettendo l'uso di dispositivi **SuperSpeed**, sebbene a una velocità inferiore, con PC dotati di porte USB 2.0. Il pinout di USB 3.0 **Type-A** e **Type-B**, riportato nella tabella 1.2, include un totale di 9 pin suddivisi in due gruppi principali. I primi 4 pin sono compatibili con USB 2.0 e includono **VBUS**, **D-**, **D+** e **GND**. I restanti 5 pin sono specifici per USB 3.0 e comprendono due coppie differenziali per la trasmissione (**TX+** e **TX-**) e la ricezione dei dati (**RX+** e **RX-**), oltre a un ulteriore pin **GND**. Nel connettore **Micro-B** è inoltre presente il pin di identificazione per USB OTG.

Rispetto a USB 2.0, la corrente massima erogabile è maggiore e pari a 900 mA, con significativi miglioramenti nella gestione dell'alimentazione, inclusa la riduzione della potenza attiva durante la trasmissione dei dati.

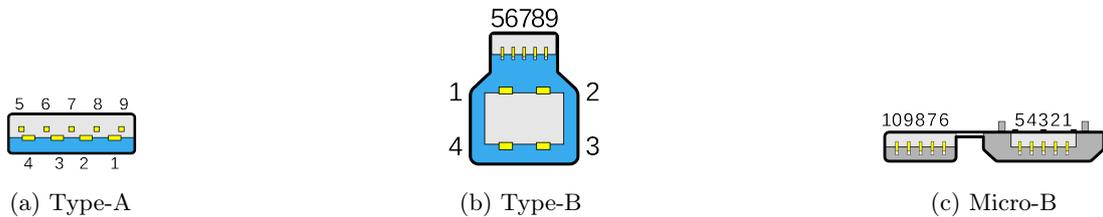


Figura 1.7: USB 3.0 Plug [2]

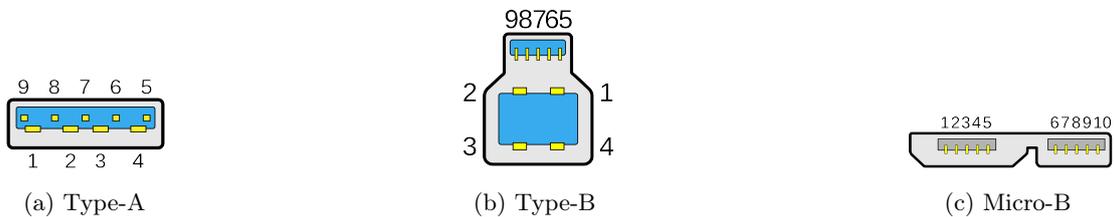


Figura 1.8: USB 3.0 Receptacle [2]

Pin	Nome segnale	Descrizione
1	VBUS	+5V
2	D-	Data-
3	D+	Data+
4	GND	Ground
5	SSTX-	SuperSpeed Transmitter -
6	SSTX+	SuperSpeed Transmitter +
7	GND	Ground
8	SSRX-	SuperSpeed Receiver -
9	SSRX+	SuperSpeed Receiver +

Tabella 1.2: Pinout connettore 3.0 Type-A, Type-B

Nel 2013 lo standard **USB 3.0** è stato aggiornato a **USB 3.1**. La principale novità di **USB 3.1** è la sua suddivisione in generazioni: **USB 3.1 Gen 1**, che mantiene la stessa velocità di trasferimento di **USB 3.0** (5 Gbps), e **USB 3.1 Gen 2**, che raddoppia la velocità massima di trasferimento fino a 10 Gbps.

Nel 2017 lo standard è stato ulteriormente aggiornato alla versione 3.2. Quest'ultima è supportata esclusivamente dal connettore **Type-C**, descritto approfonditamente nel capitolo successivo, ed è suddivisa in tre generazioni in base alla velocità massima raggiunta: **USB 3.2 Gen 1** fino a 5 Gbps, **USB 3.2 Gen 2** fino a 10 Gbps e **USB 3.2 Gen 2x2** fino a 20 Gbps.

#### 1.2.4 USB 4

**USB 4** è stato rilasciato nell'agosto del 2019, introducendo numerose innovazioni e miglioramenti. La principale novità è la velocità massima di trasferimento dei dati, che arriva fino a 40 Gbps, ideale per applicazioni che richiedono alta larghezza di banda, come video 4K e 8K. Basato sulle specifiche del protocollo **Thunderbolt 3**, **USB 4** supporta la trasmissione simultanea di dati e video, consentendo di collegare monitor esterni ad alta risoluzione. Inoltre, **USB 4** migliora l'efficienza energetica, permettendo ai dispositivi di entrare in modalità di risparmio energetico quando non sono in uso. **USB 4** è anche retrocompatibile con i dispositivi **USB 3.2** e **USB 2.0**.

#### 1.2.5 Confronto velocità e connettori USB

La tabella 1.3 riassume le velocità massime raggiunte dalle varie versioni del protocollo **USB**, mentre la tabella 1.4 mostra le connessioni supportate tra i diversi tipi di **Receptacle** e **Plug USB**.

Nome	Versione	Velocità
Low speed	USB 1.1	1.5 Mbps
Full speed	USB 1.1	12 Mbps
High speed	USB 2.0	480 Mbps
SuperSpeed	USB 3.0	5 Gbps
SuperSpeed+	USB 3.1 Gen2	10 Gbps
SuperSpeed+20	USB 3.2 Gen2x2	20 Gbps
SuperSpeed++	USB 4.0	40 Gbps

Tabella 1.3: Velocità massima supportata dai protocolli USB

## 1.2. PROTOCOLLI DATI USB

Plug	USB A	USB 3.0 A SS	USB B	USB 3.0 B SS	USB Mini-A	USB Mini-B	USB Micro-A	USB Micro-B	USB 3.0 Micro-B	USB-C
Receptacle										
	Yes	Only non-SuperSpeed	No	No	No	No	No	No	No	No
	Only non-SuperSpeed	Yes	No	No	No	No	No	No	No	No
	No	No	Yes	No	No	No	No	No	No	No
	No	No	Only non-SuperSpeed	Yes	No	No	No	No	No	No
	No	No	No	No	Yes	No	No	No	No	No
	No	No	No	No	Deprecated	Deprecated	No	No	No	No
	No	No	No	No	No	Yes	No	No	No	No
	No	No	No	No	No	No	Yes	Yes	No	No
	No	Yes	No	No						
	No	Only non-SuperSpeed	Yes	No						
	No	No	No	Yes						

Tabella 1.4: Connessioni supportate USB [2]

## 1.3 USB Power Delivery

USB Power Delivery (PD) è un protocollo avanzato che consente la negoziazione dinamica di tensione e corrente tra dispositivi, permettendo di erogare, con le ultime versioni del protocollo (fig. 1.9), fino a 240 W di potenza. Tramite USB PD, coppie di porte direttamente collegate negoziano tensione, corrente e direzione dell'alimentazione attraverso il cavo USB. Viene utilizzato un canale di comunicazione in half-duplex a 300 Kb/s che utilizza una codifica denominata Biphase Mark Coding (BMC). Quest'ultima rappresenta una modifica della codifica Manchester, in cui ad ogni zero corrisponde una transizione e ad ogni uno corrispondono due transizioni.

La struttura di un messaggio USB PD è composta da diverse sezioni:

- Il Preamble consiste in una sequenza di 64 bit utilizzata per sincronizzare trasmettitore e ricevitore.
- Lo Start Of Packet (SOP) indica l'inizio del messaggio.
- L'Header contiene informazioni sul tipo di messaggio, sul ruolo della porta, sull'identificatore del messaggio e sul numero di oggetti presenti nel payload.
- Il Payload contiene i dati effettivi del messaggio. Può includere uno o più oggetti dati (Data Object), come ad esempio: la capacità della sorgente, la richiesta di una specifica configurazione di alimentazione o i messaggi definiti dal venditore per funzionalità specifiche.
- Il Cyclic Redundancy Check (CRC) è un campo di 32 bit utilizzato per verificare l'integrità dei dati. Se il CRC non corrisponde, il messaggio viene considerato corrotto.
- L'End Of Packet (EOP) indica la fine del messaggio.

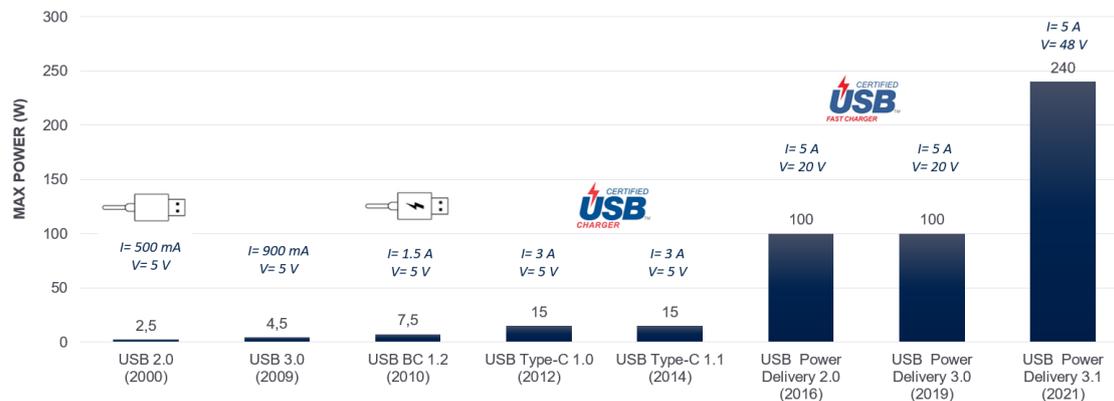


Figura 1.9: Evoluzione Power Delivery

# Capitolo 2

## USB Type-C

USB Type-C nasce dalla necessità di avere un connettore compatto, sottile e reversibile per la miniaturizzazione dei dispositivi di nuova generazione. In molti casi, i nuovi dispositivi hanno raggiunto un punto in cui i connettori USB tradizionali limitano l'innovazione, principalmente a causa delle dimensioni relativamente grandi e della scarsa robustezza dei connettori Type-A e Type-B.

Il connettore USB Type-C ha avuto inoltre un impatto positivo sull'ambiente; grazie alla standardizzazione dei cavi e alla riduzione del numero di cavi e alimentatori necessari, si è ottenuta una riduzione dei rifiuti elettronici. Inoltre, rispetto ai precedenti connettori, è stato reso più resistente e durevole, contribuendo così a una riduzione dei rifiuti derivanti da sostituzioni frequenti.

La specifica [3] è stata rilasciata nel 2014 e le caratteristiche principali sono le seguenti:

- **Nuovi Receptacle e Plug:** il nuovo connettore Type-C è più piccolo rispetto ai precedenti connettori Type-A e Type-B. Inoltre, migliora la facilità d'uso essendo inseribile in entrambe le direzioni.
- Rilevamento dei dispositivi e configurazione dell'interfaccia.
- Supporto del protocollo USB 4 e tutte le versioni precedenti.
- Fornisce supporto nativo fino a 15 W (fino a 3 A a 5 V), estendibile fino a 240 W (fino a 5 A a 48 V) con USB PD.
- Supporta diverse modalità alternative (**Alt Mode**), che permettono di trasmettere segnali video come **HDMI** e **DisplayPort**, consentendo il collegamento di monitor esterni e altri dispositivi di visualizzazione e alimentazione.

USB Type-C rappresenta quindi una soluzione versatile che combina potenza, trasferimento dati e gestione avanzata dell'alimentazione in un unico standard, facendo di USB Type-C il punto di intersezione tra il dominio della potenza, il dominio dei dati e il dominio USB PD.

## 2.1 USB Type-C Plug e Receptacle

Non tutti i dispositivi necessitano del set completo di funzionalità offerto da USB Type-C; ad esempio, alcuni potrebbero richiedere solo alimentazione e trasferimento dati di base, senza la necessità di alte velocità o del USB PD. Per questo motivo, esistono cavi USB Type-C che non includono tutti i pin della versione completa (Full-Featured). Questi cavi semplificati sono più economici e sufficienti per applicazioni meno complesse, garantendo la compatibilità con l'interfaccia USB Type-C.

La caratteristica distintiva del connettore USB Type-C è la sua reversibilità, che consente di inserirlo in qualsiasi orientamento. La presenza di pin duplicati su entrambi i lati del connettore (A e B) permette questa funzionalità, garantendo che le connessioni necessarie siano sempre disponibili. In figura 2.1 sono riportati Plug e Receptacle del USB Type-C Full Feature, la tabella 2.1 riassume l'elenco dei segnali utilizzati nei connettori USB Type-C.

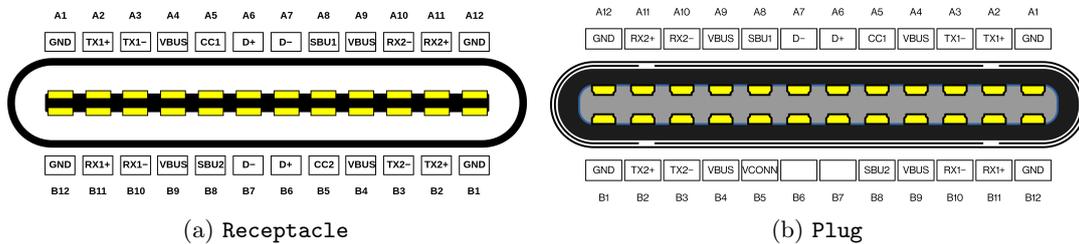


Figura 2.1: USB Type-C Full-Feature

Pin	Nome	Descrizione	Pin	Nome	Descrizione
A1	GND	Ground	B1	GND	Ground
A2	TX1+	SS Transmission +	B2	TX2+	SS Transmission +
A3	TX1-	SS Transmission -	B3	TX2-	SS Transmission -
A4	VBUS	Bus Power	B4	VBUS	Bus Power
A5	CC1	Configuration Channel	B5	CC2	Configuration Channel
A6	D+	Data+	B6	D+	Data+
A7	D-	Data-	B7	D-	Data-
A8	SBU1	Side Band Use	B8	SBU2	Side Band Use
A9	VBUS	Bus Power	B9	VBUS	Bus Power
A10	RX2-	SS Receiver -	B10	RX1-	SS Receiver -
A11	RX2+	SS Receiver +	B11	RX1+	SS Receiver +
A12	GND	Ground	B12	GND	Ground

Tabella 2.1: Pin di connessione per USB Type-C

### 2.1.1 Descrizione Pin

#### USB 3.2/USB4 Pin

- I pin TX1+, TX1- e TX2+, TX2- formano due coppie differenziali per la trasmissione ad alta velocità dei dati secondo i protocolli USB 3.2 e USB 4.
- Analogamente, i pin RX1+, RX1- e RX2+, RX2- sono altre due coppie differenziali utilizzate per la ricezione.

## USB 2.0 Pin

- I pin D+, D- sono necessari per implementare il protocollo USB 2.0. Sono supportate tutte e tre le modalità (LS, FS, HS). La presenza di due coppie di pin D+ e D- nel **Receptacle** garantisce che la connessione USB 2.0 funzioni correttamente indipendentemente dall'orientamento del **Plug**.

## Sideband Pin

- In modalità USB4, i pin SBU1 e SBU2 sono utilizzati come canale laterale (Sideband channel) per la comunicazione e sono mappati rispettivamente a SBTX e SBRX. La mappatura di SBTX e SBRX ai pin SBU1 e SBU2 viene invertita se il plug è inserito in modo opposto. Il Sideband Channel viene utilizzato per trasmettere informazioni di controllo tra i dispositivi collegati o per supportare modalità alternative come la trasmissione di segnali video o audio.

Quando il connettore USB Type-C opera in modalità USB 3.2 o USB 2.0, i pin SBU1 e SBU2 devono essere lasciati in circuito aperto o avere una debole resistenza di pull-down verso massa.

## Pin di Configurazione

- I pin Configuration Channel (CC) vengono utilizzati per rilevare il collegamento delle porte USB, stabilire l'orientamento del cavo, configurare il VBUS con la corrente definita da USB Type-C e stabilire i ruoli delle due porte collegate.

Inoltre, quando è supportato il protocollo USB PD, il pin CC connesso viene utilizzato per la comunicazione Biphase Mark Coding (BMC).

Sono presenti due pin CC1 e CC2 nel **Receptacle**, ma solo un pin CC è collegato attraverso il cavo per stabilire l'orientamento e la comunicazione. Nel caso di cavi alimentati (**Powered Cable**), che contengono elettronica all'interno del plug, l'altro pin CC viene riutilizzato per fornire la tensione VCONN, cioè come l'alimentazione dei **Plug**.

## Pin di Alimentazione

- VBUS fornisce alimentazione elettrica ai dispositivi collegati.
- VCONN viene applicato al pin CC inutilizzato per fornire alimentazione al **Plug**.
- Ground (GND) fornisce un riferimento di massa comune per tutti i segnali del connettore USB Type-C.

## 2.2 Processo di configurazione

### 2.2.1 Connessioni funzionali tra Sources e Sinks

Plug e Receptacle di USB Type-C non differenziano i ruoli di **Source** e **Sink** in base alla forma del cavo, due **Receptacle Type-C** possono essere collegate insieme attraverso un cavo indipendentemente dal ruolo. La tabella 2.2 riassume il risultato della connessione tra porte **Source**, **Sink** e **DRP**.

	<b>Source</b>	<b>Sink</b>	<b>DRP</b>
<b>Source</b>	Non funzionale	Funzionale	Funzionale
<b>Sink</b>	Funzionale	Non funzionale	Funzionale
<b>DRP</b>	Funzionale	Funzionale	Funzionale

Tabella 2.2: Interconnessioni USB Type-C

Le connessioni **Source-Source** e **Sink-Sink** non danneggiano le porte, poiché **VBUS** e **VCONN** non vengono applicati, tuttavia tali connessioni non risultano funzionali. Nel caso di due **DRP**, la risoluzione dei ruoli può avvenire automaticamente o manualmente.

Per stabilire una connessione valida tra **Source** e **Sink**, è necessario analizzare le terminazioni delle linee **CC**. Il **Source** espone due resistenze di Pull-Up  $R_p$  nelle sue terminazioni, mentre il **Sink** espone due resistenze di Pull-Down  $R_d$ .

La figura 2.2 mostra il modello equivalente di una connessione valida tra **Source** e **Sink**.

Un **Powered Cable** espone la resistenza  $R_a$  in uno dei pin **CC**, lasciando l'altro open.

Esistono due combinazioni speciali: quando entrambe le linee sono a  $R_d$ , si connettono accessori di debug; mentre se entrambe le linee sono a  $R_a$ , si è in modalità **Liquid Corrosion Mitigation**. Quest'ultima modalità è utilizzata per prevenire danni al connettore causati da eventuali liquidi.

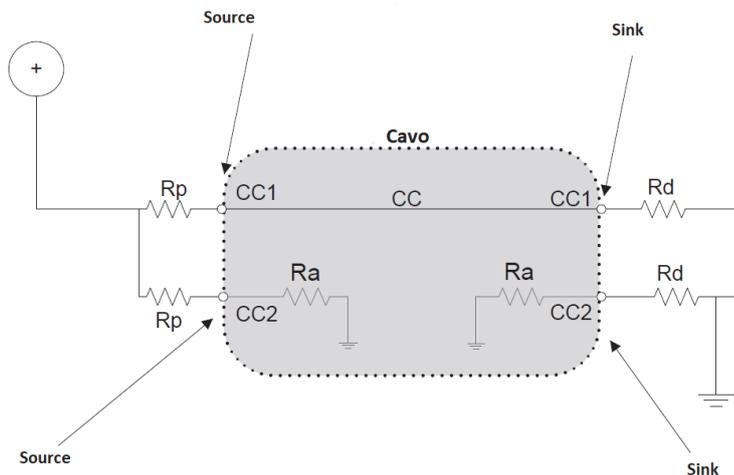


Figura 2.2: Connessione Source-Sink [4]

Se il **Source** rileva una tensione minore rispetto alla tensione di circuito aperto, allora significa che vi è una connessione. Una volta che il **Sink** è alimentato, monitora le linee **CC1** e **CC2**. La linea **CC** che viene portata ad una tensione più alta, rispetto alla tensione di

riferimento, dalla resistenza  $R_p$  del **Source** indica l'orientamento del plug.

Non appena il **Source** rileva che un **Sink** è connesso e l'orientamento è stato determinato, applica tensione al **VBUS**. Dopo una connessione, il **Source** continua a monitorare la linea **CC** per rilevare un'eventuale disconnessione. Se quest'ultima viene rilevata, il **Source** rimuove la tensione applicata a **VBUS** e **VCONN**, resetta la configurazione dell'interfaccia e cerca di rilevare una nuova connessione.

La tabella 2.3 riporta lo stato della connessione in funzione delle terminazioni delle linee **CC** dal punto di vista del **Source**.

<b>CC1</b>	<b>CC2</b>	<b>Stato</b>
Open	Open	Nessuna connessione
$R_d$	Open	Sink connesso
Open	$R_d$	
Open	$R_a$	Powered Cable senza Sink
$R_a$	Open	
$R_d$	$R_a$	Powered Cable con Sink
$R_a$	$R_d$	
$R_d$	$R_d$	Accessori per il Debug connessi
$R_a$	$R_a$	Liquid Corrosion Mitigation

Tabella 2.3: Stato connessione dal punto di vista del **Source**

Per ogni stato della tabella 2.3 si ha un diverso comportamento del **Source** e del **Sink**, riassunti nella tabella 2.4.

<b>Stato</b>	<b>Comportamento del Source</b>	<b>Comportamento del Sink</b>
Nessuna Connessione	Monitora <b>CC</b> per rilevare una connessione. Non applica <b>VBUS</b> .	Monitora <b>VBUS</b> per rilevare una connessione.
Sink connesso	Monitora <b>CC</b> per rilevare una disconnessione. Applica <b>VBUS</b> .	Monitora <b>CC</b> per rilevare l'orientamento. Monitora <b>VBUS</b> per rilevare una disconnessione.
Powered Cable senza Sink	Monitora <b>CC</b> per rilevare una connessione. Non applica <b>VBUS</b> .	Monitora <b>VBUS</b> per rilevare una connessione.
Powered Cable con Sink	Monitora <b>CC</b> per rilevare una disconnessione. Applica <b>VBUS</b> e <b>VCONN</b> .	Monitora <b>CC</b> per rilevare l'orientamento. Monitora <b>VBUS</b> per rilevare una disconnessione.
Accessori per il Debug connessi	Monitora <b>CC</b> per rilevare una disconnessione. Rinconfigura in modalità debug.	Monitora <b>VBUS</b> per rilevare una disconnessione. Rinconfigura in modalità debug.
Liquid Corrosion Mitigation	Monitora <b>CC</b> per rilevare una disconnessione.	Monitora <b>VBUS</b> per rilevare una disconnessione.

Tabella 2.4: Comportamento del **Source** e del **Sink** in base allo stato

### 2.2.2 Configurazione della potenza

Tutti i dispositivi basati su USB Type-C devono supportare la corrente standard definita da USB Type-C, ma possono anche implementare altri metodi di alimentazione definiti dalle specifiche USB. Nella negoziazione della potenza esiste un preciso ordine di precedenza che deve essere rispettato. Se viene negoziato un contratto di potenza, il dispositivo è tenuto a seguire tale contratto, ignorando qualsiasi altro accordo di alimentazione.

La tabella 2.5 riporta le modalità di erogazione della potenza, ordinate in base alla priorità (la modalità elencata nella riga superiore ha priorità rispetto a quella inferiore), insieme ai relativi valori di tensione e corrente massima.

Modalità	Tensione nominale	Corrente massima
USB Power Delivery	Configurabile fino a 48V	5A
USB Type-C @ 3.0A	5V	3A
USB Type-C @ 1.5A	5V	1.5A
USB 3.2	5V	900mA
USB 2.0	5V	500mA

Tabella 2.5: Ordine di precedenza delle modalità di erogazione della potenza

Il **Source** espone  $R_p$  sulle linee CC1 e CC2, monitorando entrambe per rilevare la presenza di un **Sink**. Il valore di  $R_p$  indica il livello di corrente massimo supportato dal **Source**. Quest'ultimo può regolare il valore di  $R_p$  per cambiare la corrente massima disponibile per il **Sink**.

# Capitolo 3

## USB Type C e Power Delivery nei Microcontrollori STM32

Con la nota applicativa AN5225 [4], STMicroelectronics fornisce una guida per l'utilizzo di USB Type-C e Power Delivery.

### 3.1 Type-C e Power Delivery controller

I microcontrollori (MCUs) e i microprocessori (MPUs) STM32 gestiscono l'interfaccia USB Type-C e USB Power Delivery utilizzando la periferica USB Type-C Power Delivery (UCPD) integrata negli STM32.[5]

L'unità UCPD comunica con il microcontrollore attraverso un'interfaccia APB. Quest'ultima è parte dell'architettura AMBA <sup>1</sup>, sviluppata da ARM per collegare periferiche a bassa larghezza di banda al core del microcontrollore.

Il controllore UCPD è integrato con un modulo analogico denominato physical layer (PHY). Quest'ultimo connette il controllore digitale ai pin CC1 e CC2 ed include i resistori  $R_p$  e  $R_d$ . La periferica può essere configurata come DFP o UFP, supportando anche il protocollo Fast Role Swap (FRS), che consente di passare rapidamente da Source a Sink e viceversa. Offre un'interfaccia di programmazione che consente al software di configurare il controllore digitale. La Direct Memory Access (DMA), permette di trasferire o ricevere le richieste del protocollo direttamente in memoria. La periferica supporta la modalità Stop che permette di ridurre il consumo di potenza. In questa modalità viene mantenuta la capacità di rilevare una connessione o una disconnessione, di rilevare i messaggi USB PD in arrivo e di segnalare eventi FRS.

#### 3.1.1 Diagramma a Blocchi

In figura 3.1 viene riportato il diagramma a blocchi della periferica UCPD.

Il PHY integrato dentro l'unità rileva i livelli di tensione sulle linee CC del Receptacle Type-C.

Il modulo Type-C controller monitora le linee CC utilizzando i segnali provenienti dai comparatori del PHY e la configurazione software dei registri del controllore per rilevare e segnalare eventuali connessioni e disconnessioni.

Il Power Delivery TX viene utilizzato per la trasmissione dei messaggi secondo le specifiche

---

<sup>1</sup>Advanced Microcontroller Bus Architecture

di USB PD, utilizzando la codifica BMC per codificare e trasmettere i dati: preambolo, SOP, dati di payload, CRC, EOP, inserendo automaticamente il gap tra i frame.

Il modulo Power Delivery RX opera da ricevitore USB PD, riconoscendo i pacchetti SOP, decodifica il flusso dei dati in arrivo, recupera il preambolo, decodifica i dati di payload, rileva EOP e verifica il CRC.

L'unità Reset e Clock Controller (RCC) gestisce il reset della periferica UCPD e fornisce il clock al kernel. Mentre il `ucpd_pclk`, fornito dal bus APB, consente l'accesso ai registri. La periferica UCPD utilizza I/O dedicati per alcune funzionalità. I pin `UCPDx_CC1` e `UCPDx_CC2` sono gli unici segnali connessi al receptacle USB Type-C. Il pin `UCPDx_FRSTX` è utilizzato da un Dual Role Port per richiedere uno scambio di ruolo, permettendo di controllare un transistor NMOS esterno che abbassa la tensione nella linea CC1 o CC2. I pin `UCPDx_DBCC1` e `UCPDx_DBCC2` sono utilizzati per indicare una condizione di batteria scarica, collegando `UCPDx_DBCC1` a `UCPDx_CC1` e `UCPDx_DBCC2` a `UCPDx_CC2`.

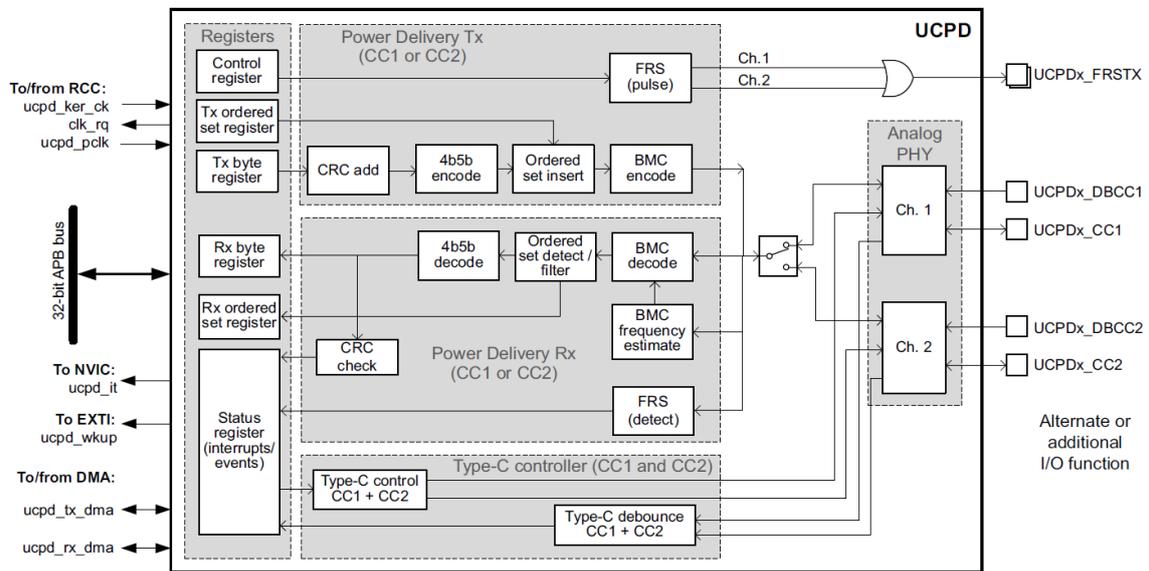


Figura 3.1: USB Type-C Power Delivery (UCPD) [5]

## 3.2 Type-C senza UCPD

Nella sezione precedente è stato discusso l'uso del microcontrollore STM32 con l'unità UCPD per sfruttare le funzionalità avanzate di USB PD.

Se tali funzionalità avanzate non sono necessarie e il microcontrollore deve essere collegato a dispositivi che supportano solo USB 2.0, è possibile utilizzare il microcontrollore senza l'unità UCPD. In questo caso, è comunque possibile utilizzare un connettore USB Type-C avvalendosi di componenti esterni. Nell'application note AN5225 [4] sono descritte le linee guida per il supporto del connettore Type-C senza l'utilizzo del UCPD.

### 3.2.1 Type-C in modalità Device

Nella modalità Device/Sink le uscite USB 2.0 del microcontrollore devono essere collegate alle due coppie di linee D+ e D- del connettore. Inoltre, per poter funzionare come Sink, è necessario presentare le resistenze  $R_d$  nelle terminazioni delle linee CC. Come mostrato in figura 3.2 due resistenze  $R_d$  da  $5.1\text{k}\Omega$  sono collegate esternamente al microcontrollore tra le linee CC e GND. Questa soluzione consente di operare come Sink con corrente massima di 500mA.

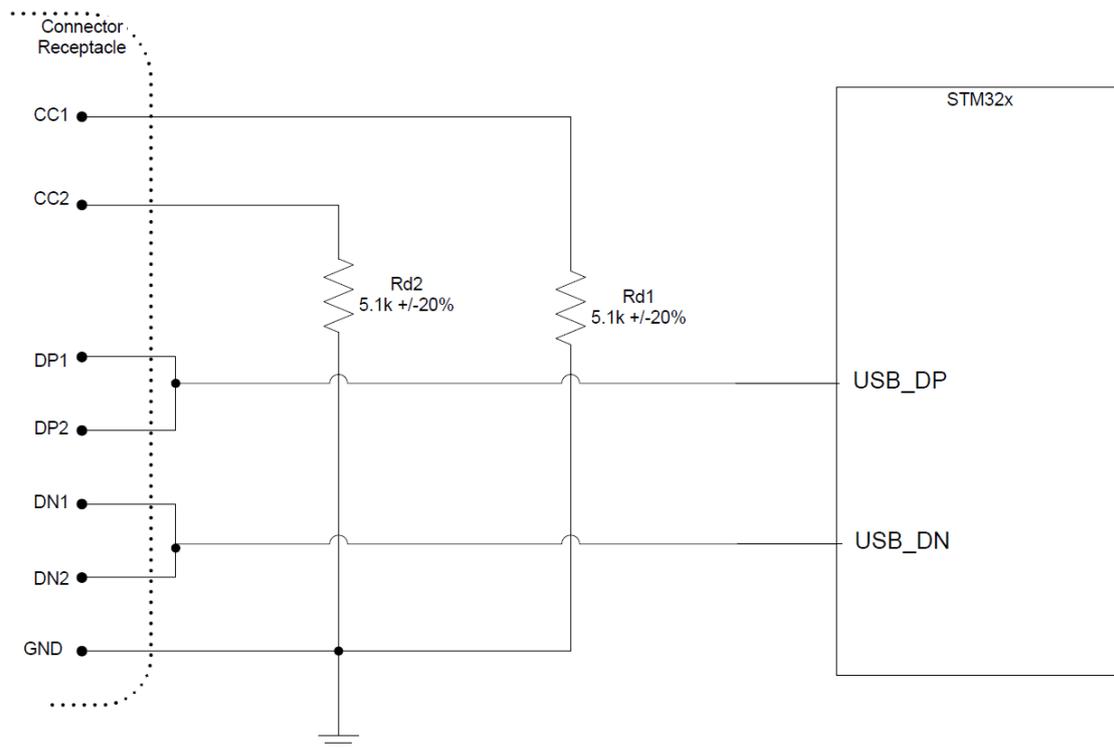


Figura 3.2: STM32 come Device [4]

### 3.2.2 Type-C in modalità Host

La figura 3.3 mostra un STM32 in modalità Host/Source.

Le uscite USB 2.0 del microcontrollore sono collegate alle coppie di linee D+ e D-. Due resistenze  $R_p$  da 56 k $\Omega$  sono esposte sulle linee CC per implementare la modalità Source e fornire la corrente di default. Per funzionare come Source, è necessario monitorare le linee CC per rilevare la connessione da parte di un Sink. A tale scopo, le linee CC sono quindi connesse a due convertitori analogico-digitale (ADC). Il microcontrollore elabora le informazioni provenienti dagli ADC e una volta rilevata la connessione, utilizza un pin ad uso generico (GPIO) per inviare il segnale di abilitazione del VBUS.

Il GPIO controlla un dispositivo esterno STMP52151 per fornire i 5V sul pin VBUS poiché il microcontrollore può fornire una tensione massima di 3.3V.

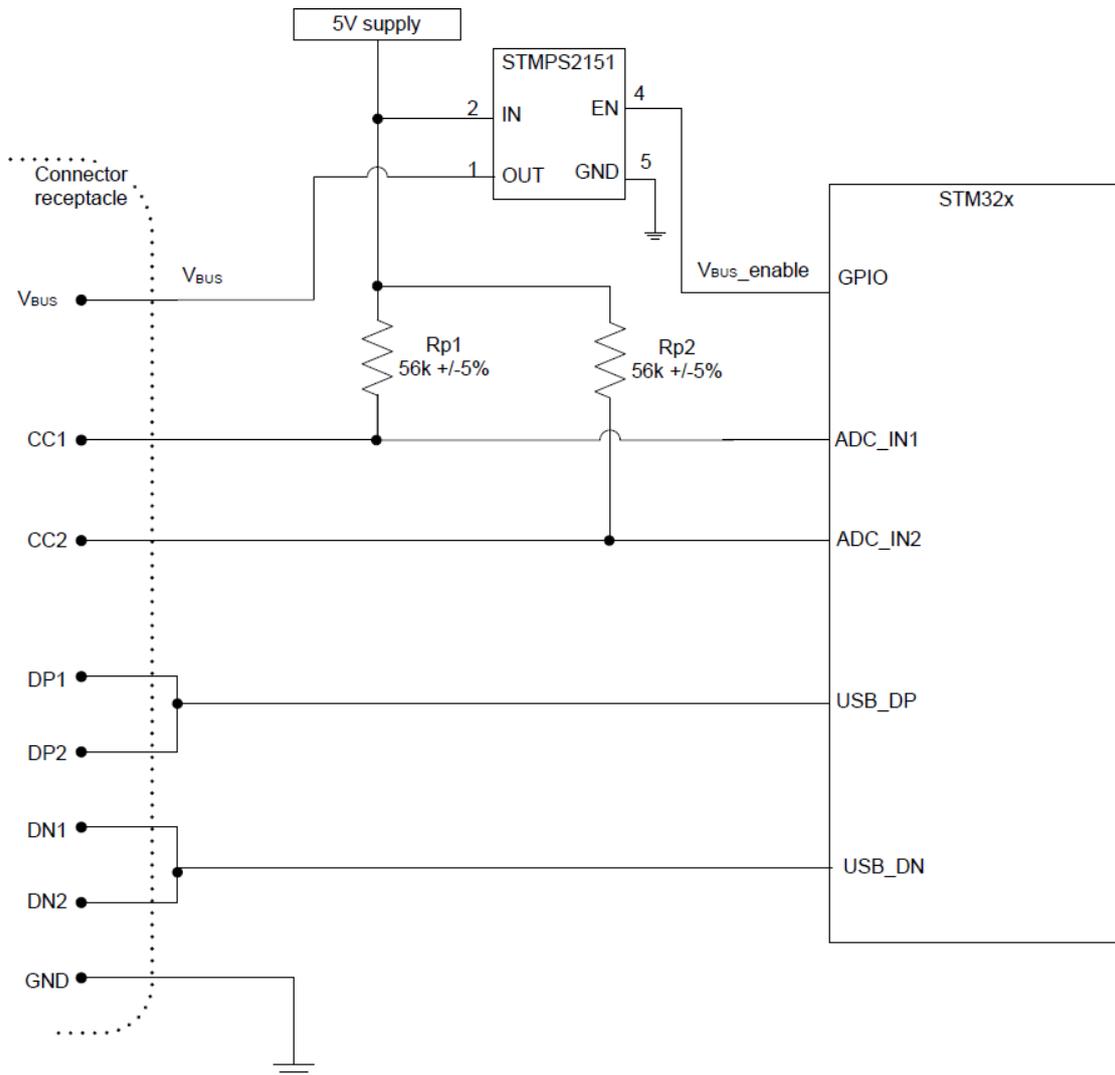


Figura 3.3: STM32 come Host [4]

### 3.2.3 Type C in modalità OTG

Come visibile dalla figura 3.4, per configurare il microcontrollore STM32 come OTG sono necessarie diverse componenti esterne a causa della possibilità di passare da **Host/Source** a **Device/Sink** e viceversa.

Un pin general purpose, **GPIO\_1**, viene collegato a due **PMOS** e due **NMOS** esterni utilizzati per configurare il microcontrollore come **DFP** o **UFP**. Un ulteriore pin general purpose, **GPIO\_2**, è utilizzato per abilitare un switch di potenza, che fornisce 5V al **VBUS** quando attivato. Un convertitore di tipo **Low Dropout Regulator (LDO)** viene impiegato per alimentare il microcontrollore. Le uscite **OTG\_FS\_DP** e **OTG\_FS\_DN** del microcontrollore sono collegate alle due coppie **D+** e **D-** del Receptacle **USB Type-C**.

Per configurare correttamente il microcontrollore **STM32**, si eseguono i seguenti passaggi:

- Inizialmente il microcontrollore viene configurato come **Sink** portando il **GPIO\_1** ad un valore alto. In questa configurazione, i **PMOS** sono spenti e gli **NMOS** accesi, così che alle terminazioni delle linee **CC** siano presenti resistenze equivalenti a  $R_d$ . Il **GPIO\_2** viene impostato a un valore basso, disabilitando il **VBUS**.
- Se viene rilevato **VBUS** dal pin **OTG\_FS\_VBUS**, l'**STM32** opera da **device/sink**.
- Se il **VBUS** non viene rilevato dopo un intervallo minimo di 200ms, il **GPIO\_1** viene portato abbassato, accendendo i **PMOS** e spegnendo gli **NMOS**. In tal caso, alle terminazioni delle linee **CC** sono presenti delle resistenze equivalenti a  $R_p$  e l'**STM32** si comporta come **Host/Source**. Il monitoraggio delle linee **CC** per rilevare la connessione di un **Sink** è effettuato da due **ADC** del microcontrollore.
- Nel caso in cui venga rilevata una connessione, il **GPIO\_2** viene abilitato attivando lo switch di potenza che fornisce la tensione necessaria al **VBUS**.



# Capitolo 4

## Sviluppo controllore Type-C only

Nel capitolo precedente sono state analizzate alcune delle soluzioni attuali fornite da **STMicroelectronics** per supportare il connettore **USB Type-C** nei microcontrollori **STM32**. Come si è visto, la periferica **UCPD** integra sia **USB PD** che **USB Type-C** in un'unica soluzione. Nel caso in cui le funzionalità di **USB PD** non siano necessarie, è possibile utilizzare microcontrollori **STM32** che non includono la periferica **UCPD**. In questo scenario, per implementare il protocollo **USB Type-C** è necessario utilizzare componenti elettronici esterni. Ad esempio, per una configurazione **Device/Sink**, la soluzione prevede l'uso di due resistori esterni. Per il funzionamento come **Host/Source**, oltre ai due resistori, è necessario impiegare due **ADC** del microcontrollore per monitorare le linee **CC** e uno switch di potenza per fornire il **VBUS**. In una configurazione **OTG**, il numero di componenti aumenta ulteriormente, poiché il microcontrollore deve essere in grado di gestire la transizione da **Host** a **Device**.

Tutte queste soluzioni comportano un aumento dei costi in termini di **Bill of Materials (BOM)**, ossia il costo dei componenti esterni da integrare nella scheda insieme al microcontrollore. Considerando che un microcontrollore può avere un costo di pochi centesimi di euro, l'impatto dei componenti esterni sui margini di guadagno può essere significativo.

L'obiettivo di questo lavoro di tesi è progettare un controllore digitale integrato che implementi esclusivamente le funzioni di **USB Type-C**, da cui il termine "**Type-C only**". Il vantaggio di questa soluzione risiede nella riduzione dell'area rispetto al controller **UCPD** e nella possibilità di utilizzare **USB Type-C** senza l'impiego di componenti esterni. Il nuovo controllore **Type-C only** consentirà di supportare le tre modalità: **Sink**, **Source** e **DRP**.

Di seguito verrà descritta la realizzazione la nuova **Intellectual Property (IP) Type-C only**. Questa **IP** si interfaccia con un due **Physical Layer** analogici (**PHY**), uno per ogni linea **CC**, ed è costituita dai seguenti moduli:

- **APB\_Interface (APB\_ITF)**
- **Kernel**
- **Sincronizzatore**
- **Wake\_Up**
- **Clock\_Divider**

## 4.1 Input e Output

L'IP è integrata con un PHY analogico, che connette il controllore digitale ai pin CC1 e CC2 e include i resistori  $R_p$  e  $R_a$ . Sono presenti due PHY, uno per ogni linea CC. Il PHY analogico, non oggetto di questa tesi, è composto da diversi comparatori che monitorano la linea CC e forniscono le corrispondenti informazioni digitali in base alla tensione rilevata.

### 4.1.1 Input e Output del PHY

Nome	Tipologia	Descrizione
CC1_CAB_DET_USB	Input	Uscita del comparatore per il rilevamento di un cavo attivo, quando l'IP è configurata come DFP e può offrire come corrente massima la corrente di default USB.
CC1_CAB_DET_1_5	Input	Uscita del comparatore per il rilevamento di un cavo attivo, quando l'IP è configurata come DFP e può offrire come corrente massima 1.5 A.
CC1_CAB_DET_3_0	Input	Uscita del comparatore per il rilevamento di un cavo attivo, quando l'IP è configurata come DFP e può offrire come corrente massima 3.0 A.
CC1_UFP_DET_1_5	Input	Uscita del comparatore per il rilevamento di una UFP, quando l'IP è configurata come DFP e può offrire come corrente massima 1.5 A.
CC1_UFP_DET_3_0	Input	Uscita del comparatore per il rilevamento di una UFP, quando l'IP è configurata come DFP e può offrire come corrente massima 3.0 A.
CC1_DFP_DET	Input	Uscita del comparatore per il rilevamento di una DFP. L'IP è configurata come UFP.
CC1_DFP_DET_USB	Input	Uscita del comparatore per il rilevamento di una DFP che può offrire come corrente massima la corrente di default USB. L'IP è configurata come UFP.
CC1_DFP_DET_1_5	Input	Uscita del comparatore per il rilevamento di una DFP che può offrire come corrente massima 1.5 A. L'IP è configurata come UFP.

Tabella 4.1: Input dell'IP provenienti dal PHY CC1

Nome	Tipologia	Descrizione
CC1_DIS_UFP_DET	Output	Disabilita i comparatori utilizzati per il rilevamento di una connessione di un UFP o di un cavo attivo quando l'IP è configurata come DFP.
CC1_DIS_DFP_DET	Output	Disabilita i comparatori utilizzati per il rilevamento di una connessione di un DFP quando l'IP è configurata come UFP.
CC1_Rd	Output	Quando è attivo, il PHY applica la $R_d$ nella linea CC1.
CC1_Rp_USB	Output	Quando è attivo, il PHY applica la $R_{p\_USBDef}$ nella linea CC1.
CC1_Rp_1_5	Output	Quando è attivo, il PHY applica la $R_{p\_1.5A}$ nella linea CC1.
CC1_Rp_3_0	Output	Quando è attivo, il PHY applica la $R_{p\_3.0A}$ nella linea CC1.

Tabella 4.2: Output dell'IP per il PHY CC1

## 4.2 Clock e Reset

Il controllore digitale **Type-C only** ha un unico segnale di reset, attivo basso, proveniente dal bus APB (PReset). Il controllore utilizza due clock: **USBClk** e **PClk**.

**USBClk** può essere diviso tramite il modulo **Clock\_Divider** e viene utilizzato dai seguenti moduli:

- Kernel
- Wake\_Up

Il **PClk** è fornito dal bus APB (**PClk**) e viene utilizzato dai registri del modulo **APB\_ITF**.

**PClk** e **UsbpdClk** sono indipendenti e la sincronizzazione tra i moduli viene effettuata tramite il modulo **Sincronizzatore**.

## 4.3 APB\_ITF

Il modulo `APB_Interface` (`APB_ITF`) contiene i registri dell'IP ed ha il compito di ricevere e gestire i segnali del bus APB. Il bus APB è costituito dal **Master**, che controlla le transizioni di lettura e scrittura, e dallo **Slave**, che risponde alle richieste del **Master**. La periferica **Type-C only** svolge il ruolo di **Slave**, ricevendo i segnali dal microcontrollore.

### 4.3.1 Lettura e scrittura APB

La tabella 4.3 descrive i segnali utilizzati per la comunicazione tra l'APB Slave dell'IP **Type-C only** e il **Master**:

Nome	Descrizione
PCLK	Clock.
nPReset	Reset asincrono attivo basso.
PAddr	Indirizzo del registro da scrivere/leggere.
PReady	Indica quando lo <b>Slave</b> è pronto per completare la transazione.
PSEL	Selezionatore del APB <b>Slave</b> .
PENABLE	Inizia il trasferimento un ciclo dopo PSEL.
PWRITE	Indica una scrittura quando il segnale presenta un valore logico alto; viceversa, indica una lettura.
PWDATA	Input dei dati dello <b>Slave</b> APB quando PWRITE è alto.
PRDATA	Output dei dati letti dello <b>Slave</b> APB quando PWRITE è basso.

Tabella 4.3: Interfaccia APB

Il processo di lettura/scrittura sul bus APB può essere suddiviso in 3 fasi: Setup, Accesso e Terminazione.

1. **Fase di Setup** - Il **Master** imposta l'indirizzo sul bus `PAddr` e attiva il segnale `PSEL` per selezionare il dispositivo **Slave**. Nel caso di lettura `PWRITE` è impostato a 0, mentre nel caso di una scrittura `PWRITE` è settato a 1. Durante questa fase `PENABLE` è a 0.
2. **Fase di Accesso** - Il **Master** porta `PENABLE` a 1 e attende che `PReady` sia attivo. Una volta che `PReady` è abilitato, se `PWRITE` è a 1 i dati su `PWDATA` vengono scritti nella periferica, viceversa il **Master** legge i dati dal segnale `PRDATA`.
3. **Fase di Terminazione** - Il **Master** disattiva `PSEL` e `PENABLE` per terminare la transizione.

La figura 4.1 mostra una lettura e una successiva scrittura all'interno del Registro `CFG1`, riportando le tre fasi del processo di lettura/scrittura. In particolare, viene scritto il campo `UCEN` del registro `CFG1` corrispondente al bit 31.

Come visto nella precedente sessione, alcuni registri possono essere scritti solo quando solo quando la periferica è disabilitata (`UCEN=0`), mentre altri possono essere modificati solo quando la periferica è abilitata. Il modulo `APB_ITF` controlla che queste condizioni siano rispettate ed inoltre verifica che non vengano effettuate scritture indesiderate nei campi riservati dei registri.

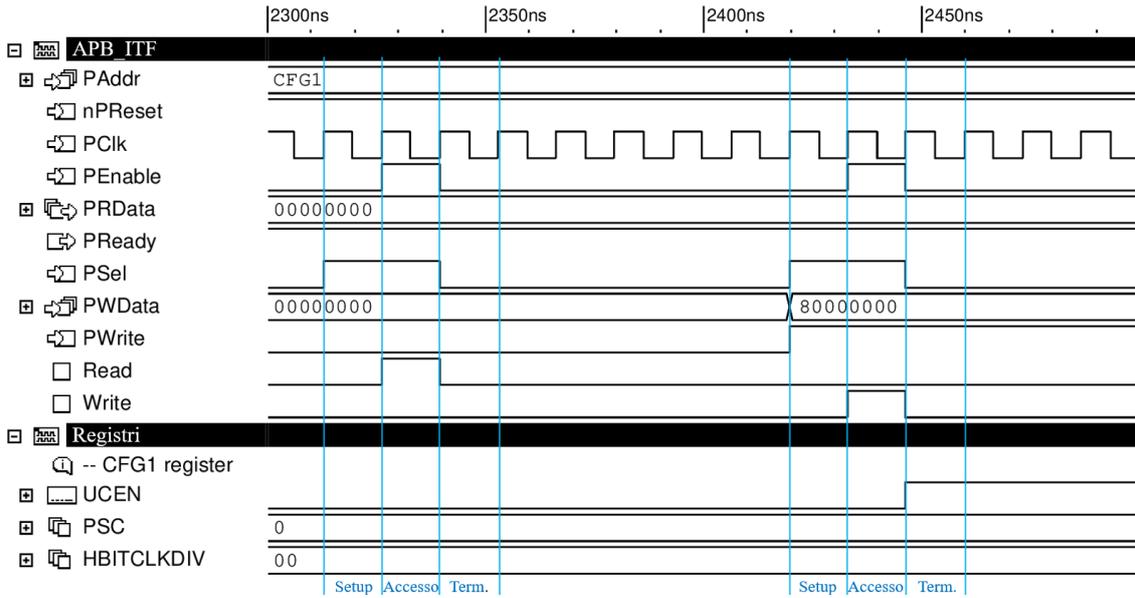


Figura 4.1: Esempio lettura e scrittura del campo UCEN

### 4.3.2 Mappa dei registri

#### Configuration register 1 (CFG1)

Offset dell'indirizzo: 0x000

Valore di Reset: 0x00000000

La scrittura in questo registro è valida solo quando l'IP è disabilitata (UCEN = '0').

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
UCEN	Res.	Res.	PSC_USBDCLK[2:0]			Res.										
rw												rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	HBITCLKDIV[5:0]															
												rw	rw	rw	rw	rw

Figura 4.2: CFG1

#### Bit 31 UCEN - USB Type-C Enable

Impostando il bit a '0', la periferica viene disabilitata e tutti i bit di controllo tornano ai loro valori di reset.

0: Disabilita

1: Abilita

Bit 19:17 **PSC[2:0]** - Prescaler USBClk

Determina il rapporto di divisione del pre-scaler per la generazione del clock USBClk.

0x0: 1 (bypass)

0x1: 2

0x2: 4

0x3: 8

0x4: 16

Bit 5:0 **HBITCLKDIV[5:0]** - Rapporto di Divisione HBITCLKEN

Determina il rapporto di divisione di USBClk utilizzato all'interno del Clock\_Divider per generare HBITCLKEN. Questo segnale è utilizzato per abilitare un contatore all'interno del modulo Kerne1, che determina se il livello di tensione sulla linea CC può essere considerato stabile.

0x00: 1 (bypass)

0x1A: 27

0x3F: 64

### Configuration register 2 (CFG2)

Offset dell'indirizzo: 0x004

Valore di Reset: 0x00000000

La scrittura in questo registro è valida solo quando l'IP è disabilitata (UCEN = '0').

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	WUPEN	FORCECLK	Res.	Res.											
												rw	rw		

Figura 4.3: CFG2

Bit 3 **WUPEN** - Enable risveglio dalla modalità Stop

0: Disabilita

1: Abilita

Bit 2 **FORCECLK** - Richiesta del clock

Consente di forzare la riattivazione del clock per uscire dallo STOP mode.

0: Non forza la richiesta del clock

1: Forza la richiesta del clock

**Configuration register 3 (CFG3)**

Offset dell'indirizzo: 0x008

Questo registro è utilizzato per il trimming via software del valore delle resistenze del PHY,  $R_p$  e  $R_d$ .

Il trimming è inizialmente effettuato dall'hardware attraverso dei valori di default. Dopo la prima scrittura nel registro, è possibile modificare i valori di regolazione delle resistenze tramite software. La scrittura in questo registro è valida solo quando UCEN = '0'.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	TRIM2_NG_CC3A0[3:0]				TRIM2_NG_CC1A5[4:0]					TRIM2_NG_CCRPD[3:0]			
			r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TRIM1_NG_CC3A0[3:0]				TRIM1_NG_CC1A5[4:0]					TRIM1_NG_CCRPD[3:0]			
			r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Figura 4.4: CFG3

Bit 28:25 **TRIM2\_NG\_CC3A0[3:0]** - Regolazione software della resistenza  $R_{p\_3.0A}$  della linea CC2.

Bit 24:20 **TRIM2\_NG\_CC1A5[4:0]** - Regolazione software della resistenza  $R_{p\_1.5A}$  della linea CC2.

Bit 19:16 **TRIM2\_NG\_CC3A0[3:0]** - Regolazione software della resistenza  $R_d$  della linea CC2.

Bit 12:9 **TRIM1\_NG\_CC3A0[3:0]** - Regolazione software della resistenza  $R_{p\_3.0A}$  della linea CC1.

Bit 8:4 **TRIM1\_NG\_CC1A5[4:0]** - Regolazione software della resistenza  $R_{p\_1.5A}$  della linea CC1.

Bit 3:0 **TRIM1\_NG\_CC3A0[3:0]** - Regolazione software della resistenza  $R_d$  della linea CC1.

**Control register (CR)**

Offset dell'indirizzo: 0x00C

Valore di Reset: 0x00000000

La scrittura in questo registro è valida solo quando UCEN = '1'.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2TCDIS	CC1TCDIS	Res.	Res.	Res.	Res.
										rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CC2VCONNEN	CC1VCONNEN	Res.	CCENABLE[1:0]		ANAMODE	ANASUBMODE[1:0]		PHYCCSEL	Res.	Res.	Res.	Res.	Res.	Res.
	rw	rw		rw	rw	rw	rw	rw	rw						

Figura 4.5: CR

Bit 21 **CC2TCDIS** - CC2 Type-C detector disable

Abilita i comparatori del PHY della linea CC2.

0: Abilita

1: Disabilita

Bit 20 **CC1TCDIS** - CC1 Type-C detector disable

Abilita i comparatori del PHY della linea CC1.

0: Abilita

1: Disabilita

Bit 14 **CC2VCONNEN** - VCONN switch enable per CC2

Abilita la VCONN sulla linea CC2.

0: Disabilitato

1: Abilitato

Bit 13 **CC1VCONNEN** - VCONN switch enable per CC1

Abilita la VCONN sulla linea CC1.

0: Disabilitato

1: Abilitato

Bit 11:10 **CCENABLE[1:0]** - Enable linea CCAbilita i PHY (resistenze  $R_p$  e  $R_d$ ) delle linee CC1 e CC2 secondo l'impostazione di ANAMODE e ANASUBMODE. Solitamente, vengono abilitati entrambi i PHY. L'abilitazione di un singolo PHY può essere utile, ad esempio, quando l'altra linea è usata per fornire la VCONN

0x0: Disabilita entrambi i PHY

0x1: Abilita il PHY di CC1

0x2: Abilita il PHY di CC2

0x3: Abilita i PHY di CC1 e CC2

Bit 9 **ANAMODE** - Analog PHY operating mode

Determina la modalità di funzionamento del PHY analogico.

0: Source

1: Sink

Bit 8:7 **ANASUBMODE[1:0]** - Analog PHY sub-mode

Determina la modalità di funzionamento del PHY analogico, in combinazione con **ANAMODE**.

Se l'IP è configurata come **Sink**, **ANASUBMODE** non è rilevante; mentre se l'IP è configurata come **Source**, **ANASUBMODE** determina il profilo di corrente massima.

0x0: Resistenze  $R_p$  disabilitate

0x1:  $R_{p\_USBDef}$

0x2:  $R_{p\_1.5A}$

0x3:  $R_{p\_3.0A}$

Bit 6 **PHYCCSEL** - Selettore linea CC1/CC2

0: Deseleziona I/O CC1

1: Deseleziona I/O CC2

La selezione dipende dall'orientamento del cavo.

### Interrupt mask register (IMR)

Offset dell'indirizzo: 0x010

Valore di Reset: 0x00000000

La scrittura in questo registro è valida solo quando l'IP è abilitata ( $UCEN = '1'$ ).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPECEVT2IE	TYPECEVT1IE	Res.													
rw	rw														

Figura 4.6: IMR

Bit 15 **TYPECEVT2IE** - TYPECEVT2 interrupt enable

Abilita la maschera dell'interrupt TYPECEVT2.

0: Interrupt disabilitato

1: Interrupt abilitato

Bit 14 **TYPECEVT1IE** - TYPECEVT1 interrupt enable

Abilita la maschera dell'interrupt TYPECEVT1.

0: Interrupt disabilitato

1: Interrupt abilitato

**Registro di stato (SR)**

Offset dell'indirizzo: 0x014

Valore di reset: 0x00000000

La scrittura in questo registro è valida solo quando UCEN = '1'.

31	30	29	28	27	26	25	24	23	22	21	20	19:18		17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TYPEC_VSTATE_CC2[1:0]		TYPEC_VSTATE_CC1[1:0]	
												r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPECEVT2	TYPECEVT1	Res.	Res.	Res.	Res.										
r	r														

Figura 4.7: SR

Bit 19:18 **TYPEC\_VSTATE\_CC2[1:0]** - Voltage Level CC2

Indica il livello di tensione sulla linea CC2 nel suo stato stazionario.

0x0: Più basso

0x1: Basso

0x2: Alto

0x3: Più alto

Bit 17:16 **TYPEC\_VSTATE\_CC1[1:0]** - Voltage Level CC1

Indica il livello di tensione sulla linea CC1 nel suo stato stazionario.

0x0: Più basso

0x1: Basso

0x2: Alto

0x3: Più alto

Bit 15 **TYPECEVT2** - Type-C event CC2Flag che indica un cambiamento del valore di **TYPEC\_VSTATE\_CC2[1:0]**, corrisponde a un nuovo evento **Type-C**. Il flag viene cancellato impostando il bit **TYPECEVT2CF** del registro **ICR**.

0: Nessun nuovo evento

1: Un nuovo evento **Type-C**Bit 14 **TYPECEVT1** - Type-C event CC1Flag che indica un cambiamento del valore di **TYPEC\_VSTATE\_CC1[1:0]**, corrisponde a un nuovo evento **Type-C**. Il flag viene cancellato impostando il bit **TYPECEVT1CF** del registro **ICR**.

0: Nessun nuovo evento

1: Un nuovo evento **Type-C**

**Interrupt clear register (ICR)**

Offset dell'indirizzo: 0x018

Valore di reset: 0x00000000

La scrittura in questo registro è valida solo quando l'IP è abilitata (UCEN = '1').

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPECEVT2CF	TYPECEVT1CF	Res.													
w	w														

Figura 4.8: ICR

Bit 15 **TYPECEVT2CF** - Type-C CC2 event clear

Il settaggio di questo bit cancella il corrispettivo evento TYPECEVT2 del SR.

Bit 14 **TYPECEVT1CF** - Type-C CC1 event clear

Il settaggio di questo bit cancella il corrispettivo evento TYPECEVT1 del SR.

## 4.4 Clock\_Divider

Nel modulo `Clock_Divider`, il clock `USBC1k` viene suddiviso tramite rapporti di divisione determinati dal prescaler (`PSC`) e dal parametro `HBITCLKDIV`, generando un segnale di abilitazione chiamato `HBITCLKEN`. Questo segnale è utilizzato per attivare un contatore all'interno del modulo `Kernel`, descritto nella sezione successiva, che determina se il livello di tensione sulla linea `CC` può essere considerato stabile.

Modificando i valori del `PSC` e del parametro `HBITCLKDIV` presenti nel modulo `APB_ITF`, è possibile variare il periodo di generazione del segnale `HBITCLKEN`. Aumentando il periodo, si incrementa il tempo necessario per considerare stabile il livello di tensione sulla linea `CC`. Questo consente di selezionare, tramite software, il tempo necessario per considerare stabile un determinato livello sulle linee `CC`.

## 4.5 Sincronizzatore

Il modulo `Sincronizzatore` gestisce la comunicazione tra il `Kernel` e `APB_ITF`.

Quando due moduli digitali, operanti a frequenze di clock diverse, devono comunicare si possono verificare problematiche legate alla temporizzazione. Uno dei problemi più rilevanti in questi contesti è la metastabilità. Quest'ultima si verifica quando un segnale in ingresso a un flip-flop cambia troppo vicino al fronte di clock attivo, impedendo al flip-flop di raggiungere uno stato stabile entro il tempo richiesto. In condizioni normali, un flip-flop si stabilisce rapidamente in uno dei due stati logici ('0' o '1'), ma in presenza di metastabilità, può rimanere in uno stato indefinito per un tempo variabile. Questo comportamento può propagarsi nella logica a valle, causando malfunzionamenti.

Il Mean Time Between Failures (**MTBF**) è una misura statistica, indica la frequenza con cui il sistema può sperimentare uno stato metastabile che non si risolve correttamente entro i limiti temporali definiti. Il **MTBF** dipende da diversi fattori, tra cui la frequenza del clock e i tempi di setup e hold, e può essere migliorato attraverso strategie di sincronizzazione che garantiscono ai segnali asincroni il tempo necessario per stabilizzarsi. Le strategie comunemente utilizzate includono l'impiego di registri di sincronizzazione a cascata e l'implementazione di moduli di sincronizzazione basati su handshake a quattro fasi.

Il modulo `Sincronizzatore` implementa un handshake a quattro fasi, le componenti principali sono le seguenti:

- Domini di clock differenti: `PC1k` e `USBC1k`
- Segnali coinvolti nell'handshake a quattro fasi:
  - Request (**REQ**): segnale inviato dal dominio sorgente per indicare che un segnale è pronto per essere trasferito.
  - Acknowledge (**ACK**): segnale inviato dal dominio destinatario per confermare che il dato o il segnale è stato ricevuto correttamente.

Per illustrare il funzionamento dell'handshake a quattro fasi nella sincronizzazione tra due domini di clock differenti, si considera il trasferimento del segnale `ANAMODE` dal dominio `PC1k` al dominio `USBC1k`:

1. Richiesta di Trasferimento - Il dominio `PC1k` inizia il trasferimento attivando il segnale `REQ`. Questo segnale indica che il dato, in questo caso il segnale `ANAMODE`, è pronto per essere inviato al dominio di clock `USBC1k`. Il segnale `REQ`, generato all'interno del dominio `PC1k`, viene sincronizzato con il dominio `USBC1k` attraverso due o più flip-flop a cascata.
2. Conferma di Ricezione - Una volta che il dominio `USBC1k` rileva il segnale `REQ` (sincronizzato nel suo dominio), campiona il segnale `ANAMODE`. Dopo aver acquisito il dato, il dominio `USBC1k` attiva il segnale `ACK` per confermare che il segnale è stato ricevuto correttamente.
3. Rilascio della Richiesta - Dopo aver ricevuto la conferma attraverso il segnale `ACK`, il dominio `PC1k` disattiva il segnale `REQ`. Questa azione segnala che il trasferimento del segnale `ANAMODE` è stato completato con successo e che il dominio `PC1k` è pronto per iniziare un nuovo ciclo di trasferimento.
4. Rilascio della Conferma - Infine, dopo aver rilevato la disattivazione del segnale `REQ`, il dominio `USBC1k` disattiva il segnale `ACK`. Questo passaggio conferma che il segnale `ANAMODE` è stato ricevuto correttamente e che il sistema è pronto per il prossimo ciclo di trasferimento.

## 4.6 Kernel

Il modulo `Kernel` è responsabile della determinazione dello stato delle linee `CC`, a partire dalle uscite del `PHY` e dall'impostazione dell'IP da parte del software. Inoltre, include una macchina a stati dedicata alla generazione degli eventi `Type-C`.

Per semplificare la trattazione, verrà illustrato solo il funzionamento relativo alla linea `CC1`, poiché la logica applicata alla linea `CC2` è identica.

Lo stato della linea `CC1` è codificato nel segnale `VSTATE_CC1`. Quest'ultimo risulta dipendere dai valori di `ANAMODE`, `ANASUBMODE`, `CCEnable` e dalle uscite dei comparatori del `PHY`. I segnali `ANAMODE`, `ANASUBMODE` e `CCEnable`, come descritto nel modulo `APB_ITF`, sono scritti dal software nel registro `CR` e sono sincroni nel dominio `APB`. Per essere utilizzati all'interno del `Kernel`, devono essere sincronizzati nel dominio `USBClk` dal Sincronizzatore.

All'interno del `Kernel` vengono generati i segnali `ANAMODE_IEvt`, `ANASUBMODE_IEvt` e `CCEnable_IEvt`. Questi segnali vengono attivati per un colpo di clock ogni volta che si verifica un cambiamento nei rispettivi segnali. La seguente tabella riporta la codifica utilizzata per `VSTATE_CC1`, `CC1_Voltage` indica il livello di tensione reale nella linea `CC1`.

VSTATE_CC1	ANAMODE	ANASUBMODE	vCC1	CC1_Voltage [V]	PHY	
11	0 (Source)	00(Disabilita)	X	X.X	-	
00		01 ( $R_{pUSB}$ )	$vR_a$	0.1	CC1_UFP_DET_1_5 = 0 CC1_CAB_DET_USB = 0	
01			$vR_d$	0.3	CC1_UFP_DET_1_5 = 0 CC1_CAB_DET_USB = 1	
10			$vR_{open}$	1.7	CC1_UFP_DET_1_5 = 1 CC1_CAB_DET_USB = 1	
00		10 ( $R_{p1.5A}$ )	$vR_a$	0.1	CC1_UFP_DET_1_5 = 0 CC1_CAB_DET_1_5 = 0	
01			$vR_d$	0.5	CC1_UFP_DET_1_5 = 0 CC1_CAB_DET_1_5 = 1	
10			$vR_{open}$	1.7	CC1_UFP_DET_1_5 = 1 CC1_CAB_DET_1_5 = 1	
00		11 ( $R_{p3.0A}$ )	$vR_a$	0.1	CC1_UFP_DET_3_0 = 0 CC1_CAB_DET_3_0 = 0	
01			$vR_d$	0.9	CC1_UFP_DET_3_0 = 0 CC1_CAB_DET_3_0 = 1	
10			$vR_{open}$	2.8	CC1_UFP_DET_3_0 = 1 CC1_CAB_DET_3_0 = 1	
00		1 (Sink)	XX	$vR_a$	0.1	CC1_DFP_DET = 0 CC1_DFP_DET_1_5 = 0 CC1_DFP_DET_USB = 0
01				$vR_{pUSB}$	0.3	CC1_DFP_DET = 1 CC1_DFP_DET_1_5 = 0 CC1_DFP_DET_USB = 0
10	$vR_{p1.5A}$			0.9	CC1_DFP_DET = 1 CC1_DFP_DET_1_5 = 0 CC1_DFP_DET_USB = 1	
11	$vR_{p3.0A}$			1.5	CC1_DFP_DET = 1 CC1_DFP_DET_1_5 = 1 CC1_DFP_DET_USB = 1	

Tabella 4.4: `VSTATE_CC1[1:0]`

Il segnale `Vstate_CC1` essendo generato a partire dalle uscite del `PHY`, è asincrono e deve essere sincronizzato nel dominio del `Kernel`. La sincronizzazione avviene mediante l'utilizzo di due flip-flop per ogni bit, generando `TypeCVStateCC1`.

I segnali `CC1_Level_00`, `CC1_Level_01`, `CC1_Level_10`, e `CC1_Level_11` vengono generati a partire dal segnale `TypeCVStateCC1` secondo la seguente tabella.

TypeCVStateCC1	CC1_Level_00	CC1_Level_01	CC1_Level_10	CC1_Level_11
00	1	0	0	0
01	0	1	0	0
10	0	0	1	0
11	0	0	0	1

Tabella 4.5: Generazione segnali `CC1_Level_xx`

I quattro segnali `CC1_Level_xx`<sup>1</sup> generano i corrispettivi segnali `CC1_Level_xx_q`<sup>2</sup> attraverso un processo sincrono.

Quando il segnale `PReset` è attivo, tutti i segnali `CC1_Level_xx_q` vengono impostati a '0'. Se durante il fronte di salita di `USBClk`, almeno uno dei segnali `ANAMODE_IEvt`, `ANASUBMODE_IEvt` o `CCEnable_IEvt` risulta attivo, tutti i segnali `CC1_Level_xx_q` vengono resettati a '0'. Viceversa, ciascun segnale `CC1_Level_xx_q` assume il valore corrente del corrispondente segnale `CC1_Level_xx`.

I segnali `CC1_Level_xx_qq`<sup>3</sup> rappresentano versioni ritardate dei segnali `CC1_Level_xx_q`.

I segnali `CC1_Level_xx_q` generano i segnali `CC1_Level_xx_IEvt`<sup>4</sup>. Questi ultimi indicano una variazione del livello di tensione nella linea `CC1`.

Un segnale `CC1_Level_xx_IEvt` assume valore pari a '1' solo quando il corrispondente segnale `CC1_Level_xx` è a '1' e il valore campionato `CC1_Level_xx_q` è a '0'.

Il contatore `CC1_Counter` genera il segnale `CC1_TCounter`. Quest'ultimo, segnala che un livello di tensione è stato rilevato per un intervallo di tempo sufficiente da essere considerato stabile. Se si verifica almeno una delle condizioni indicati nella tabella 4.6, il contatore viene resettato al valore "110" (6 in decimale). Durante il fronte di salita di `USBClk`, se il segnale `HBitClkEn` è attivo, il contatore viene decrementato. Il contatore si ferma quando arriva a "000" e attende di essere ripristinato al valore "110". Quando il contatore raggiunge il valore "001" (1 in decimale), il segnale `CC1_TCounter` viene attivato.

I segnali `CC1_Level_xx_Stable`<sup>5</sup> segnalano che un determinato livello di tensione è stato rilevato sulla linea `CC1` per un intervallo di tempo sufficiente. Se si verifica almeno una delle condizioni specificate nella tabella 4.6, tutti i segnali `CC1_Level_xx_Stable` vengono impostati a '0'. Durante il fronte di salita di `USBClk`, se il segnale `CC1_TCounter` è attivo, viene settato il segnale `CC1_Level_xx_Stable` corrispondente a `CC1_Level_xx`.

<sup>1</sup>`CC1_Level_00`, `CC1_Level_01`, `CC1_Level_10` e `CC1_Level_11`

<sup>2</sup>`CC1_Level_00_q`, `CC1_Level_01_q`, `CC1_Level_10_q` e `CC1_Level_11_q`

<sup>3</sup>`CC1_Level_00_qq`, `CC1_Level_01_qq`, `CC1_Level_10_qq` e `CC1_Level_11_qq`

<sup>4</sup>`CC1_Level_00_IEvt`, `CC1_Level_01_IEvt`, `CC1_Level_10_IEvt` e `CC1_Level_11_IEvt`

<sup>5</sup>`CC1_Level_00_Stable`, `CC1_Level_01_Stable`, `CC1_Level_10_Stable` e `CC1_Level_11_Stable`

PRreset = '0'
CC1_Level_00_IEvt = '1'
CC1_Level_01_IEvt = '1'
CC1_Level_10_IEvt = '1'
CC1_Level_11_IEvt = '1'
ANAMODE_IEvt = '1'
ANASUBMODE_IEvt = '1'
CCEnable_IEvt = '0'
CCEnable = "00"
ANAMODE = '0' e ANASUBMODE = "00"

Tabella 4.6: Condizioni di reset del contatore CC1\_Counter, della macchina a stati 1\_FSM e dei segnali CC1LvlxxSeen

#### 4.6.1 CC1 Finite State Machine

Per monitorare lo stato della linea CC1, viene utilizzata la macchina a stati (FSM) denominata CC1 FSM, che fa uso dei segnali precedentemente descritti.

La CC1 FSM è caratterizzata dai seguenti stati:

- **RESET** - Questo stato rappresenta lo stato iniziale della macchina. La FSM entra in questo stato ogni volta che il segnale di PRreset è attivo o si verifica una delle condizioni della tabella 4.6. La macchina permane in questo stato fintanto che si verificano le seguenti condizioni: CC1ENABLE = '0' o ANAMODE = '0' e ANASUBMODE = "00". Quando queste condizioni non sono verificate, la FSM transita dallo stato RESET allo stato UNSTABLE.
- **UNSTABLE** - In questo stato, la FSM monitora i segnali CC1\_Level\_xx\_Stable per stabilire se la linea CC1 ha raggiunto uno stato di tensione stabile. Lo stato UNSTABLE funge da stato intermedio tra i vari stati di tensione stabile. Quando uno dei segnali CC1\_Level\_xx\_Stable diventa attivo, la macchina esce dallo stato UNSTABLE e passa allo stato STABLE\_XX corrispondente.
- **STABLE\_00** - Questo stato rappresenta il livello di tensione più basso sulla linea CC1. La FSM entra in questo stato quando il livello di tensione sulla linea CC1 è stabile e corrisponde al livello più basso. La FSM esce dallo stato STABLE\_00 e passa allo stato UNSTABLE se il segnale corrispondente CC1\_Level\_00\_qq viene disattivato.
- **STABLE\_01** - Rappresenta il secondo livello di tensione sulla linea CC1. L'FSM giunge in questo stato, dallo stato UNSTABLE, quando CC1\_Level\_01\_Stable è attivo, ed esce da questo stato quando CC1\_Level\_01\_qq = '0'.
- **STABLE\_10** - Questo stato rappresenta il terzo livello di tensione sulla linea CC1. La FSM entra in questo stato quando il livello di tensione sulla linea CC1 è stabile e corrisponde al livello più basso. La FSM esce dallo stato STABLE\_10 e passa allo stato UNSTABLE se il segnale corrispondente CC1\_Level\_10\_qq viene disattivato.
- **STABLE\_11** - Analogamente agli altri stati STABLE\_XX, precedentemente descritti, si giunge in questo stato quando il corrispettivo CC1\_Level\_11\_Stable risulta attivo e si esce da questo stato quando CC1\_Level\_11\_qq = '0'.

## Output FSM

Il segnale TYPECEVT1 viene generato ogni volta che la macchina a stati entra in un nuovo stato stabile STABLE\_XX, cioè quando lo stato attuale è UNSTABLE e lo stato successivo è uno dei STABLE\_XX. Il segnale TYPECEVT1 viene sincronizzato dal Sincronizzatore e successivamente scritto all'interno del registro SR del APB\_ITF. Inoltre, ad ogni evento TYPECEVT1, il segnale TypeCVStateCC1, anch'esso sincronizzato, viene scritto all'interno dello SR.

## FSM Diagram

La figura 4.9 mostra l'evoluzione degli stati della macchina a stati. Per non appesantire la figura, i collegamenti di ogni stato con lo stato di RESET non sono mostrati esplicitamente. Come descritto in precedenza, la macchina a stati ritorna allo stato di RESET, indipendentemente dallo stato attuale, ogni qualvolta il segnale PReset è attivo o se, durante il fronte di salita, si verifica almeno una delle condizioni della tabella 4.6.

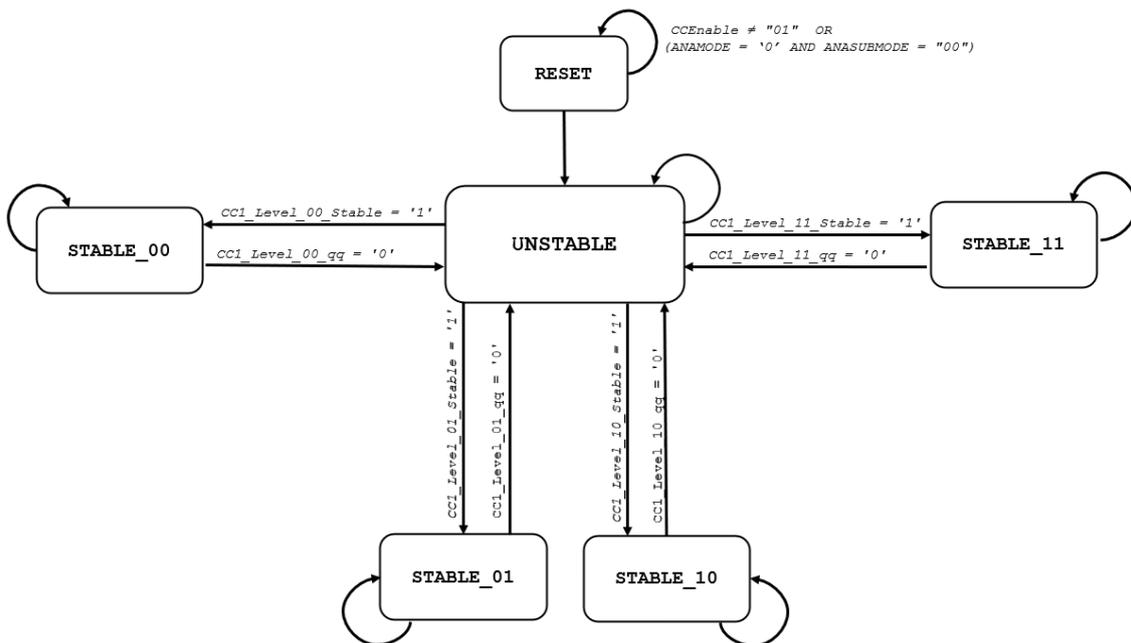


Figura 4.9: Transizione stati FSM

## 4.7 Wake-Up

Il modulo Wake-Up gestisce la linea di wake-up asincrona, che è responsabile del risveglio di entrambi i clock del IP dalla modalita STOP. Quest'ultima è progettata per ottimizzare il consumo energetico disattivando i clock della periferica finché non si verificano eventi sulle linee CC che attivano il risveglio del microcontrollore.

Per consentire il risveglio, è necessario che il bit WUPEN nel registro CFG2 sia abilitato a '1'.

# Capitolo 5

## Simulazioni e Sintesi

### 5.1 Simulazioni

Il controllore digitale **Type-C Only** è stato simulato per valutarne le principali funzionalità. Le simulazioni, eseguite con **Cadence Xcelium**, si sono concentrate inizialmente sulla verifica dei registri, controllando che il reset, le operazioni di lettura e scrittura, e la gestione dei bit riservati avvenissero correttamente.

Successivamente, il controllore è stato testato in modalità **Source**, variando le condizioni operative per simulare le diverse correnti massime erogabili e le tensioni sulle linee **CC1** e **CC2**. Sono state testate configurazioni con una sola linea abilitata (**CC1** o **CC2**) e con entrambe attive. Lo stesso approccio è stato utilizzato per le simulazioni in modalità **Sink**, con l'obiettivo di verificare il rispetto delle specifiche previste.

#### 5.1.1 Test dei Registri

Questo test verifica il valore di reset ed esegue accessi di lettura/scrittura ai seguenti registri: **CFG1**, **CFG2**, **CFG3**, **CR**.

##### Reset dei registri

In questa fase del test, si verifica il comportamento dei registri durante il reset. L'obiettivo è assicurarsi che, all'attivazione del segnale di reset, i registri assumano i valori corretti definiti nella mappa dei registri. Come mostrato nella figura 5.1, quando il segnale **PRreset** è basso, i registri assumono i valori di default.

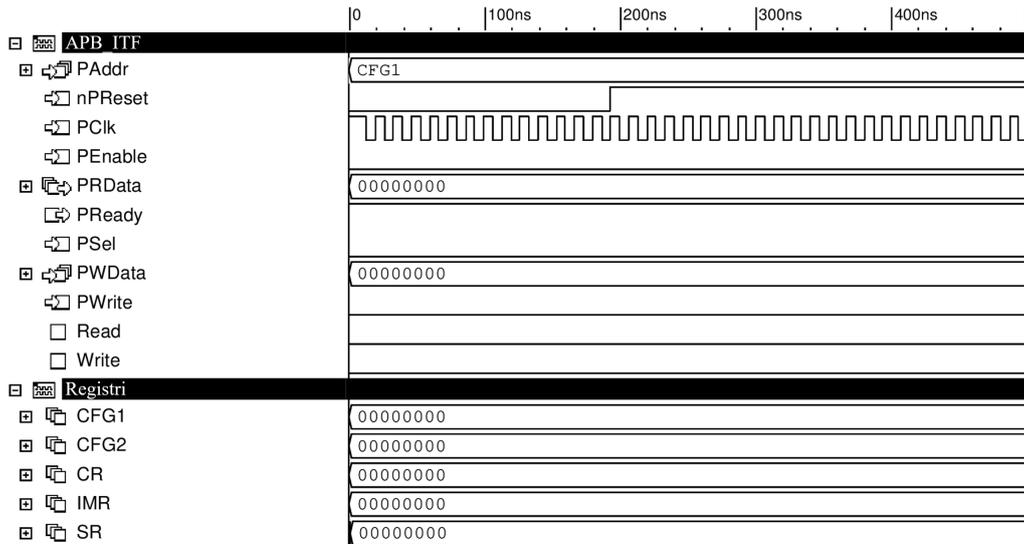


Figura 5.1: Reset dei registri

### Test del registro CFG1

In questa parte del test viene effettuata una scrittura nel registro CFG1, attivando la periferica e impostando PSC e HBITCLKDIV. Successivamente, viene letto lo stesso registro per verificare il dato precedentemente scritto. Come illustrato nella figura sottostante, l'APB\_ITF scrive soltanto nei bit non riservati del registro. Durante la fase di scrittura, con il segnale di Write impostato a '1', il valore di PWDData è 0xFFFFFFFF. Tuttavia, in fase di lettura, con il segnale di Read impostato a '1', si osserva un valore differente nel segnale PRData, pari a 0x800E003F. Questo risultato indica che solo i bit relativi a UCEN, PSC e HBITCLKDIV sono stati impostati a '1', mentre gli altri bit, essendo riservati, non sono stati modificati.

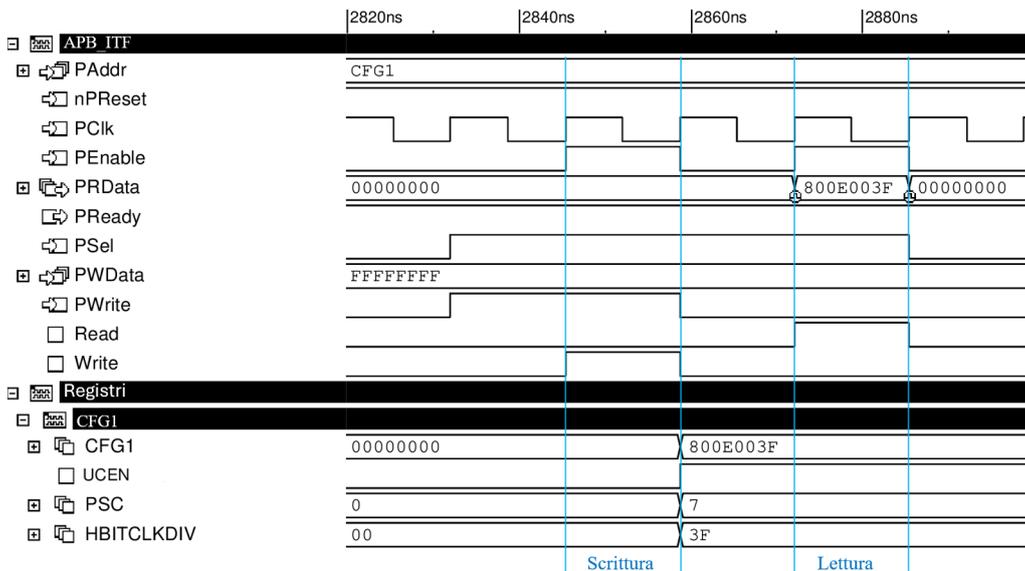


Figura 5.2: Test scrittura/lettura CFG1

### Test del registro CFG2

Analogamente, viene effettuata una scrittura nel secondo registro di configurazione (CFG2), seguita da una lettura per verificare i dati scritti. Anche in questo caso, si osserva che l'APB\_ITF esclude i bit riservati dalla scrittura, aggiornando solo i bit specificati nella mappa dei registri.

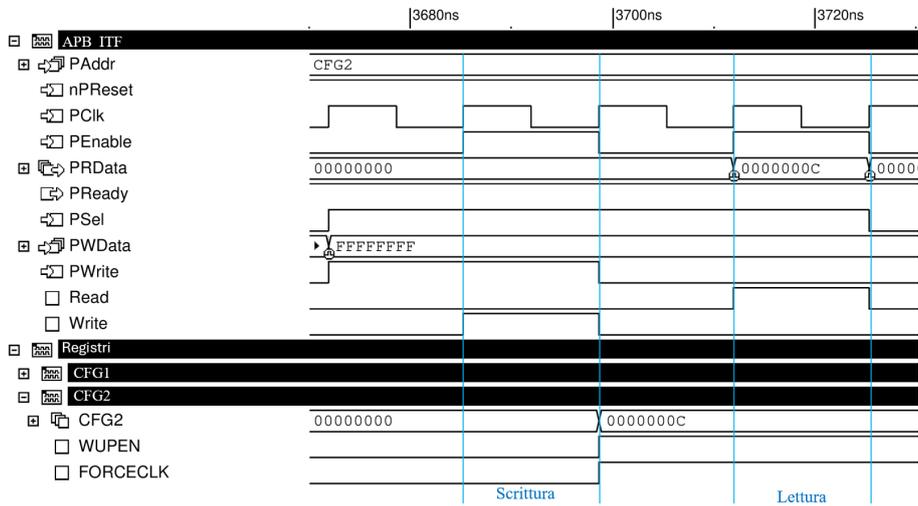


Figura 5.3: Test scrittura/lettura CFG2

### Test del registro CFG3

Il registro CFG3 è utilizzato per la regolazione software delle resistenze del PHY,  $R_p$  e  $R_d$ . Inizialmente, il valore del registro è determinato dall'hardware, come mostrato nella simulazione, dove il valore iniziale è  $0x14A814A8$ . Successivamente, viene eseguita una scrittura software con il valore  $0xFFFFFFFF$ . Come previsto, nella simulazione si osserva che solo i bit non riservati vengono aggiornati, ottenendo così il valore  $0x1FFF1FFF$ .

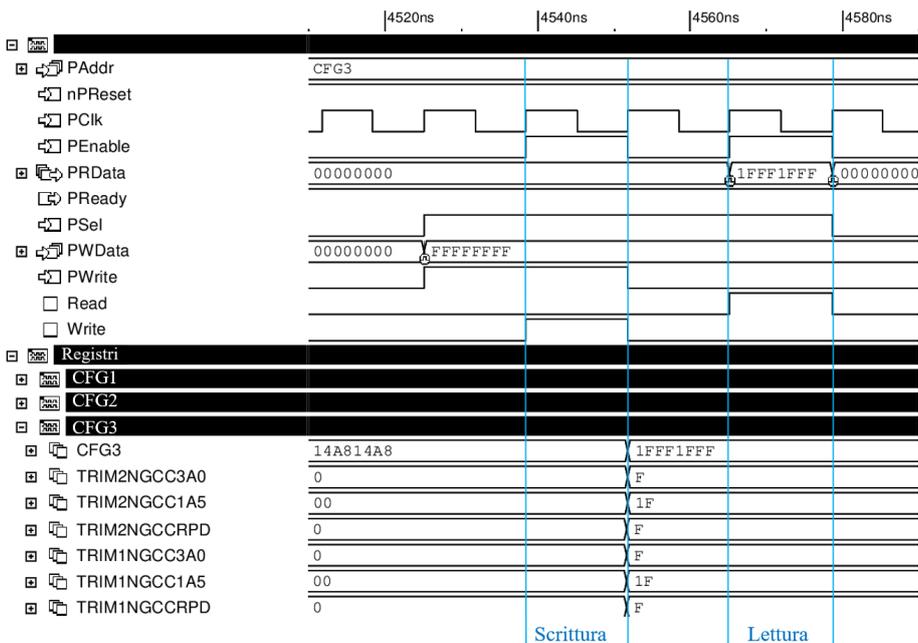


Figura 5.4: Test scrittura/lettura CFG3

### Test del registro CR

A differenza dei precedenti registri, è possibile scrivere all'interno del registro CR solo se l'IP è attiva. Come illustrato nella figura 5.5, quando il segnale UCEN è disattivato, la scrittura nel registro CR non viene effettuata.

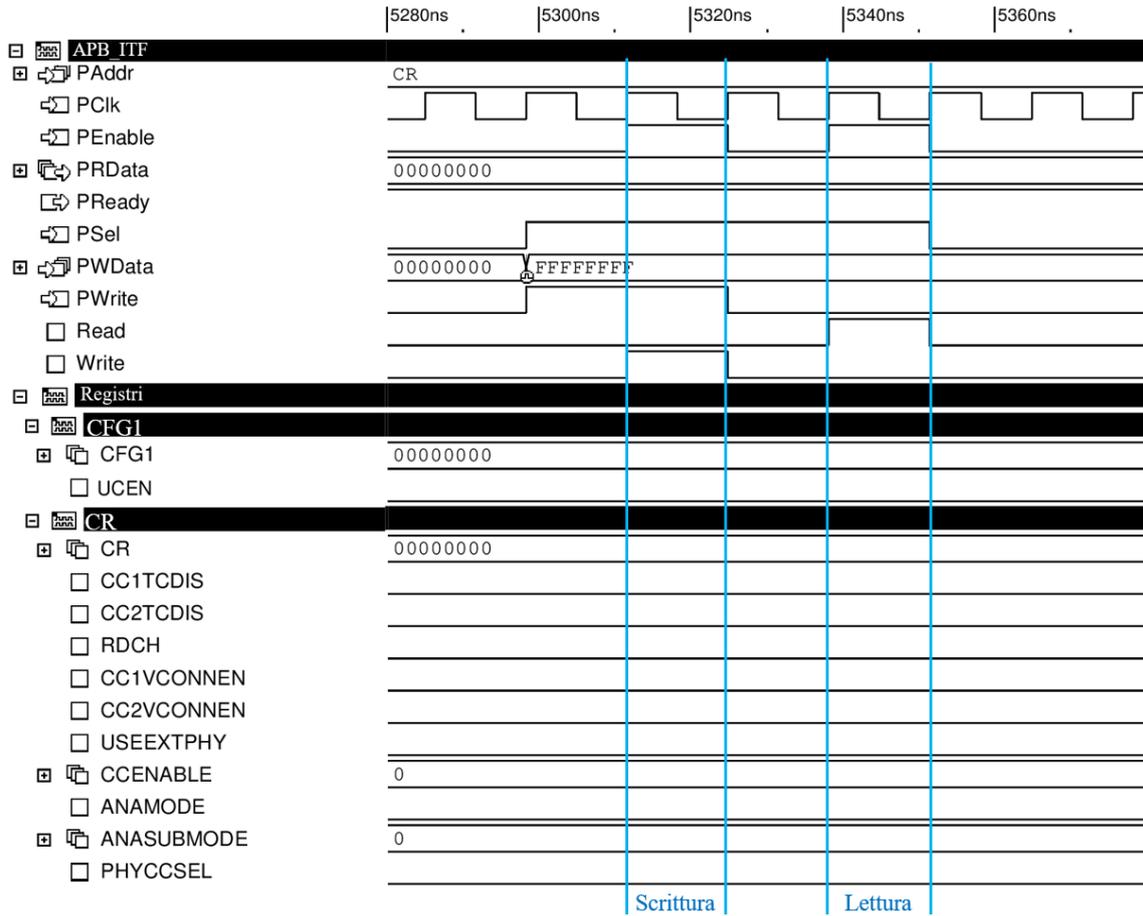


Figura 5.5: Scrittura CR con IP disabilitata

Il test prosegue dimostrando che abilitando l'IP, il registro CR viene scritto correttamente. La figura 5.6 mostra chiaramente che, una volta attivata l'IP tramite il segnale UCEN, la scrittura nel registro CR avviene come previsto, garantendo che le modifiche siano applicate solo quando il dispositivo è correttamente abilitato.

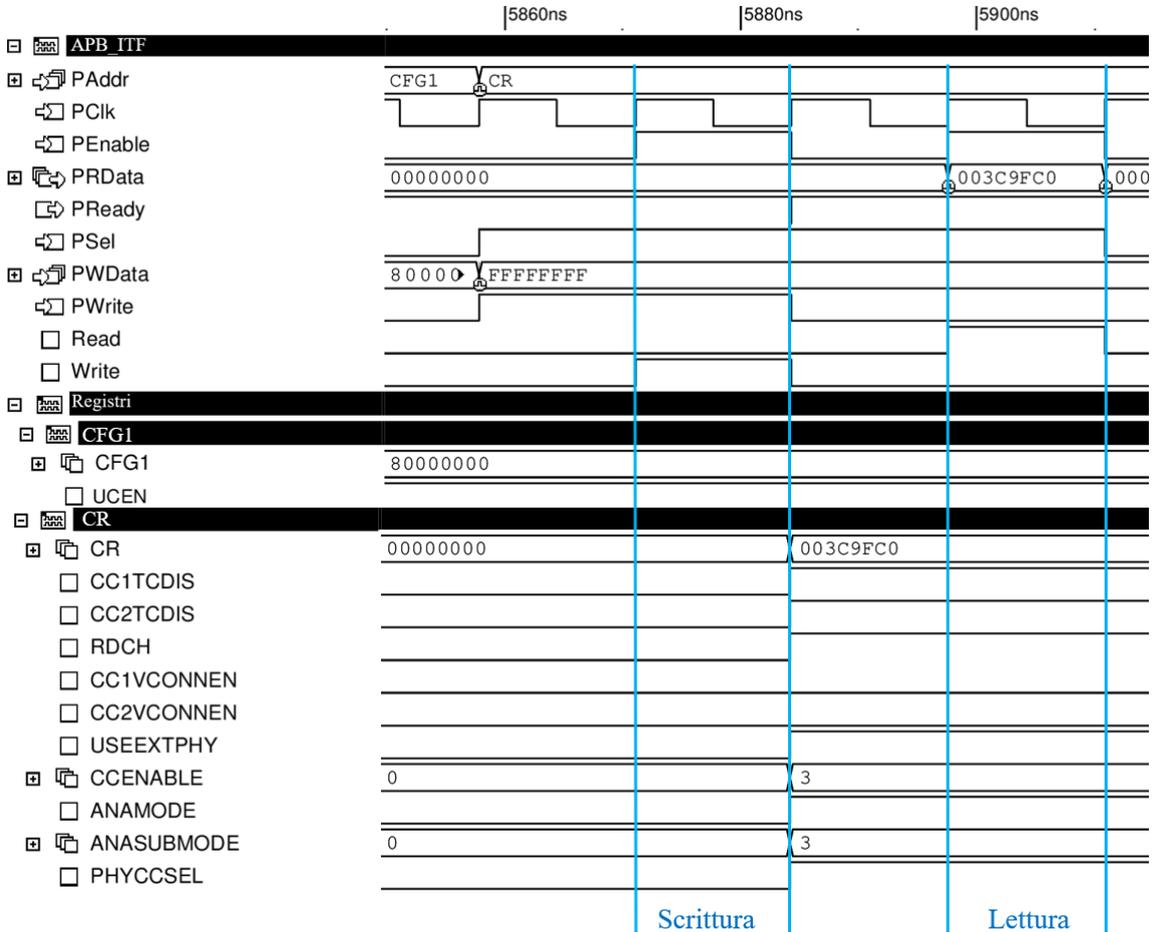


Figura 5.6: Scrittura CR con IP abilitata

## 5.1.2 Test Source

### Descrizione Test

In questa simulazione, l'IP è stata configurata come **Source** (ANAMODE=0).

La configurazione viene testata imponendo tutte le combinazioni di ANASUBMODE, CCEnable e i vari livelli di tensione sulla linea CC1/CC2, come definiti nella tabella 4.4.

Per non appesantire la trattazione, vengono riportate solo alcune parti del test, tralasciando i test sulla linea CC2, poiché valgono le medesime considerazioni.

**ANASUBMODE = "01", CC1\_Voltage = 0.6V**

Il test inizia con due scritture nel registro CFG1 (fig. 5.7). Nella prima scrittura vengono configurati PSC e HBITCLKDIV, mentre nella seconda viene abilitata l'IP attraverso l'attivazione del segnale UCEN.

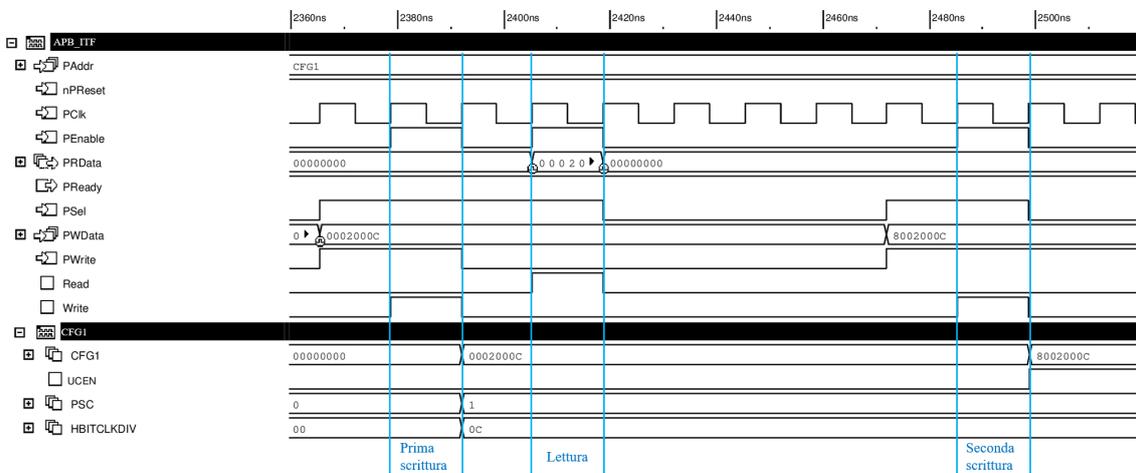


Figura 5.7: Configurazione PSC, HBITCLKDIV e abilitazione IP (UCEN)

Successivamente, viene effettuata una scrittura all'interno del CR (fig. 5.8). La linea CC1 viene abilitata, l'IP è configurata come **Source** (ANAMODE = '1') e ANASUBMODE è impostato a "01", corrispondente alla corrente USB di default.

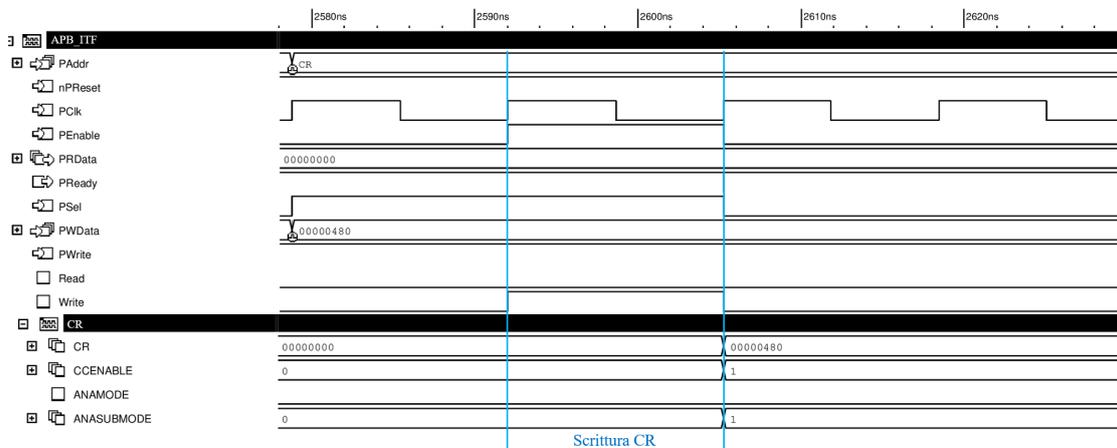


Figura 5.8: Scrittura CR: ANAMODE = '0', ANASUBMODE = "01" e CCEnable = "01"

Il segnale `CC1_DIS_UFP_DET` viene settato a '0' (fig. 5.9), abilitando i comparatori responsabili del rilevamento di un Sink o di un cavo attivo. Il test simula l'utilizzo dell'IP come Source, quindi da questo momento in poi il segnale `CC1_DIS_UFP_DET` rimarrà costante per tutta la durata del test.

Inoltre, viene abilitato il segnale `CC1_Rp_USB`, un segnale di input al PHY utilizzato per collegare la resistenza  $R_{p\_USBDef}$  alla terminazione della linea CC1. Infine, si osserva il cambiamento delle uscite dei comparatori del PHY, con l'attivazione dei segnali `CC1_CAB_DET_USB` e `CC1_CAB_DET_1_5`.

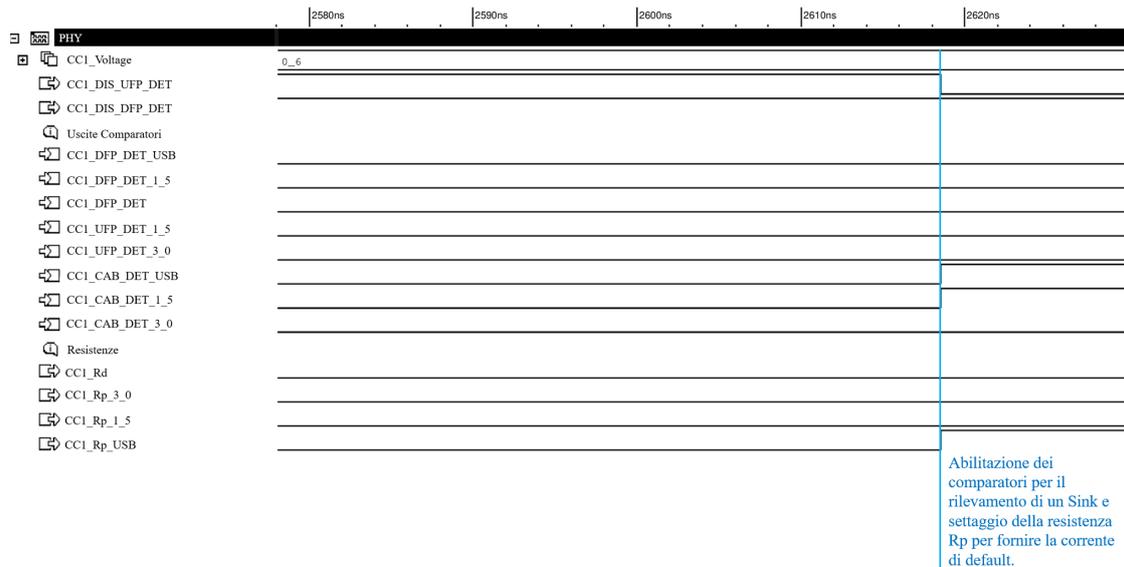


Figura 5.9: Input e Output PHY dopo la scrittura del CR

Come visibile dalla simulazione (fig. 5.10), `TypeCVSTATECC1` assume il valore pari a "01"<sup>1</sup>. Di conseguenza, il segnale `CC1_Level_01` viene attivato, seguito nei cicli successivi dai segnali `CC1_Level_01_q` e `CC1_Level_01_qq`, e il contatore `CC1_Counter` viene resettato a "110" (6). La macchina a stati passa dallo stato di RESET allo stato UNSTABLE. Poiché il livello di tensione sulla linea CC1 rimane invariato e non vengono generati eventi interni relativi ai segnali `ANAMODE`, `ANASUBMODE` e `CCENABLE`, il contatore decresce ogni volta che il segnale `HBitClkEn` è attivo. Quando il contatore arriva a "001" (1), si attiva il segnale `CC1_TCounter`, e il segnale `CC1_Level_01_Stable` viene attivato. A questo punto, la macchina a stati passa dallo stato UNSTABLE allo stato STABLE\_01, generando l'evento `CC1Evt`.

<sup>1</sup>`ANAMODE = '0'`, `ANASUBMODE = "01"`, `CCEnable = "01"`, `CC1_UFP_DET_1_5 = '0'` e `CC1_CAB_DET_USB = '1'`

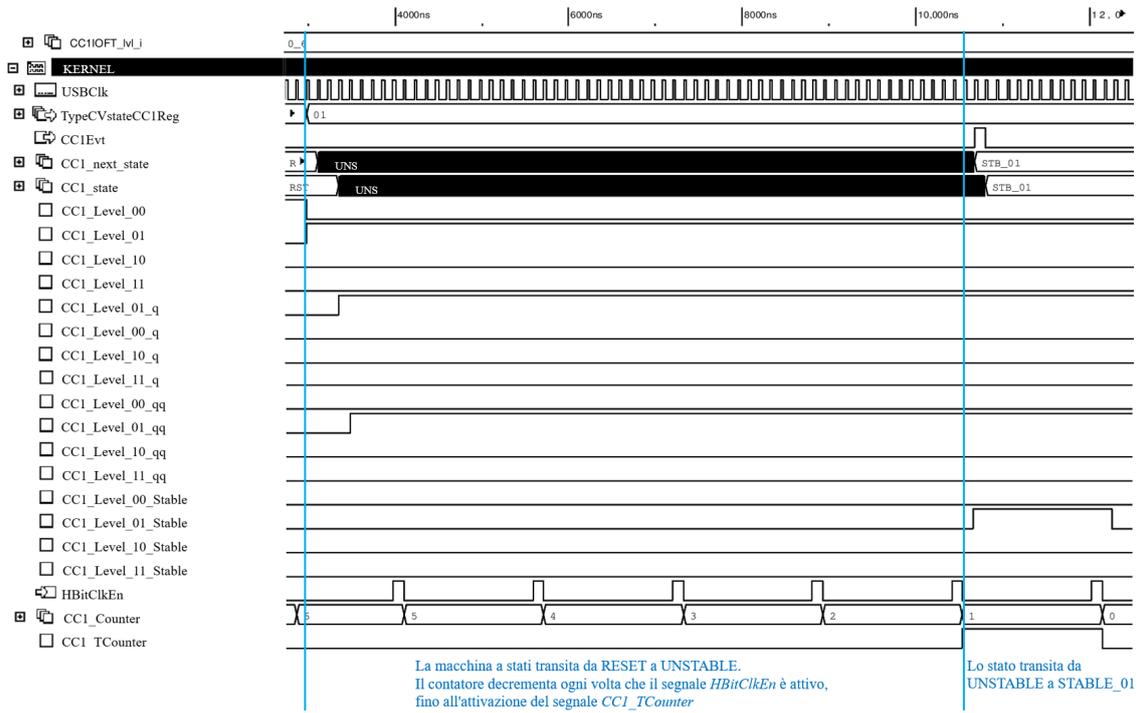


Figura 5.10: FSM STABLE\_01

Infine, come visibile in figura 5.11, l'evento **TYPECEVT1** e il valore attuale di tensione nella linea **CC1**, **TypeCVStateCC1**, vengono scritti all'interno del registro **SR**.

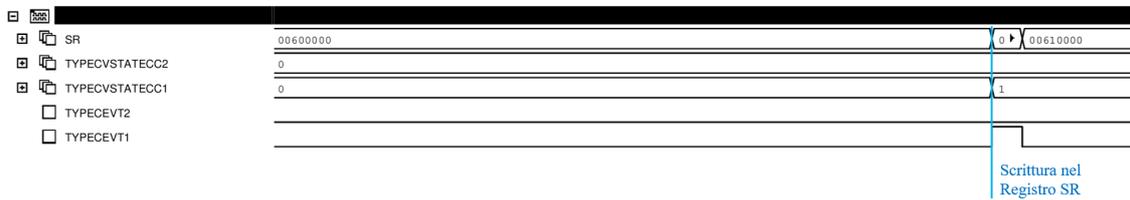


Figura 5.11: Scrittura TYPECEVT1 e TypeCVStateCC1 in SR

**ANASUBMODE = "10", CC1\_Voltage = 1.7V**

La simulazione, in figura 5.12, continua con il cambiamento del valore di ANASUBMODE da "01" (corrispondente alla corrente massima USB default) a "10" (corrente massima di 1.5A).

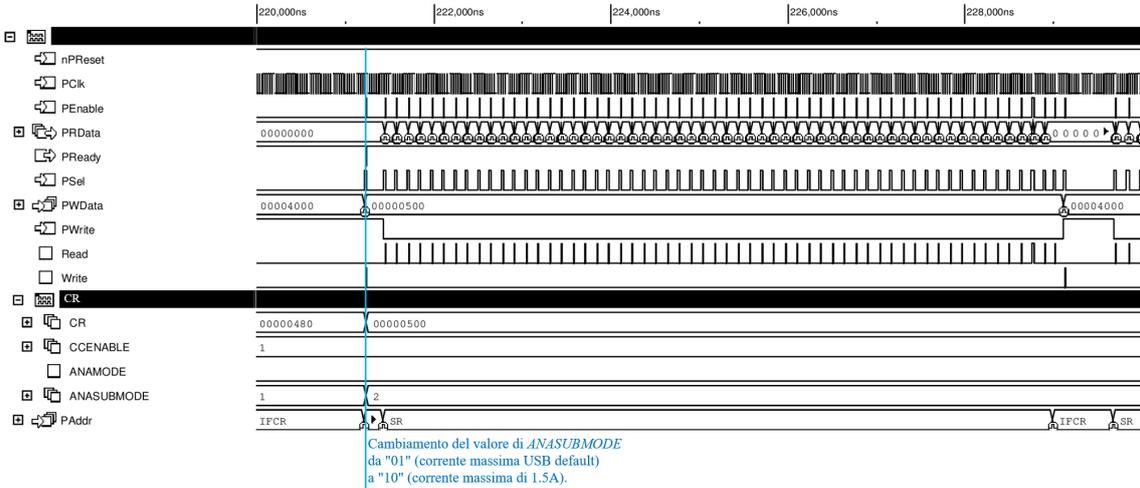


Figura 5.12: Passaggio ANASUBMODE da "01" a "10"

Di conseguenza, con il cambiamento del valore di ANASUBMODE, viene disattivato il segnale CC1\_Rp\_USB e attivato il segnale CC1\_Rp\_1\_5. Questo cambiamento comporta che il PHY, da questo momento in poi, applicherà la resistenza  $R_{p\_1.5A}$  alla terminazione della linea CC1 (fig. 5.13).

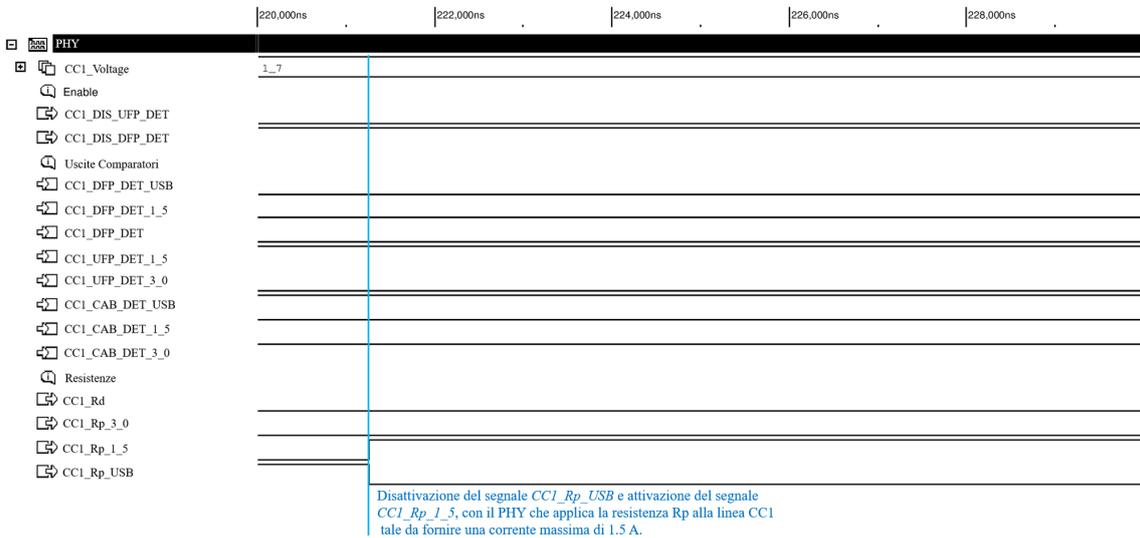


Figura 5.13: PHY ANASUBMODE = "10"

Come visibile in figura 5.14, essendo cambiato il valore di ANASUBMODE, la FSM transita nello stato UNSTABLE. Dopo diversi colpi di clock, poichè il livello di tensione sulla linea CC1 è stabilmente a 1.7 V, la macchina a stati passa allo stato STABLE\_10, secondo quando visto precedentemente.

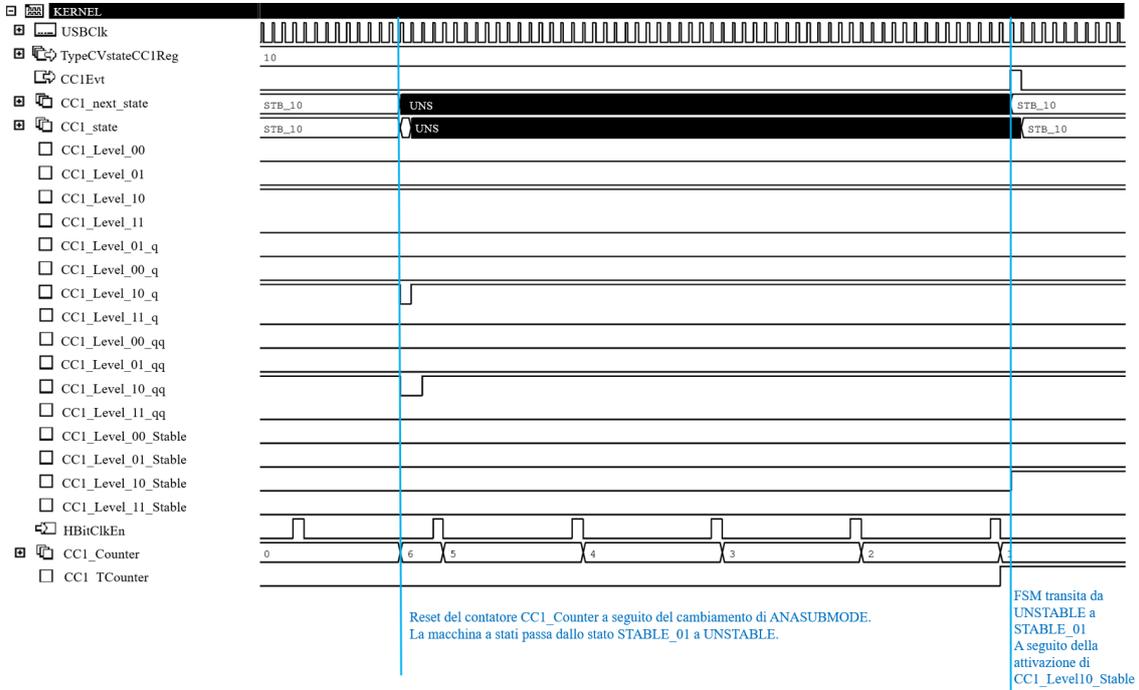


Figura 5.14: FSM dopo il cambiamento di ANASUBMODE

### 5.1.3 Test Sink

#### Descrizione Test

In questo test, l'IP è stata configurata come Sink ( $ANAMODE = '1'$ ). La configurazione viene testata imponendo tutte le combinazioni di  $ANASUBMODE$ ,  $CCEnable$  e tutti i livelli di tensione sulla linea  $CC1/CC2$ , come definiti nella tabella 4.4.

Per non appesantire la trattazione, viene riportata solo una parte del test, mostrando esclusivamente i segnali relativi alla linea  $CC2$ .

#### $CC2\_Voltage = 1.5V$

Il test inizia con la configurazione dell'IP in modalità Sink con  $ANAMODE = '1'$ . Inoltre,  $CCEnable$  viene impostato a "10", abilitando solo la linea  $CC2$  (fig. 5.15). Il valore di  $ANASUBMODE$  risulta ininfluente quando l'IP è configurata come Sink.

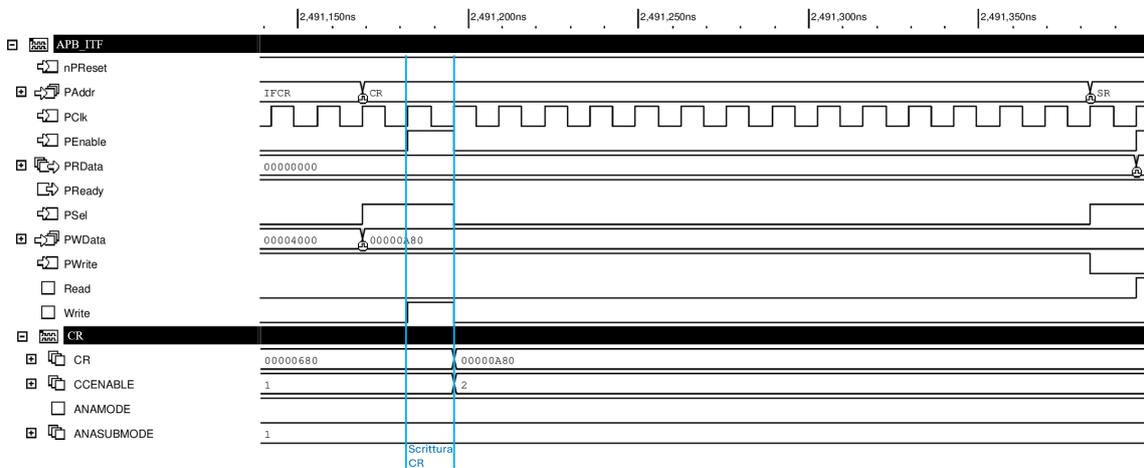


Figura 5.15: Scrittura all'interno del CR per configurare l'IP come Sink e abilitare la linea  $CC2$

Il segnale  $CC2\_DIS\_DFP\_DET$  viene settato a '0', abilitando i comparatori responsabili del rilevamento di un Source. Inoltre, viene abilitato il segnale  $CC2\_Rd$ , utilizzato come input al PHY per collegare la resistenza  $R_d$  alla terminazione della linea  $CC2$ . Il test simula l'utilizzo dell'IP come Sink, per cui, da questo momento, i segnali  $CC2\_DIS\_DFP\_DET$  e  $CC2\_Rd$  rimarranno attivi per tutta la durata del test. A seguito degli 1.5V presenti nella linea  $CC2$ , dopo l'impostazione degli input del PHY, si osserva (fig. 5.16) il cambiamento delle uscite dei comparatori con l'attivazione dei segnali  $CC2\_DFP\_DET$ ,  $CC2\_DFP\_DET\_USB$  e  $CC2\_DFP\_DET\_1\_5$ .

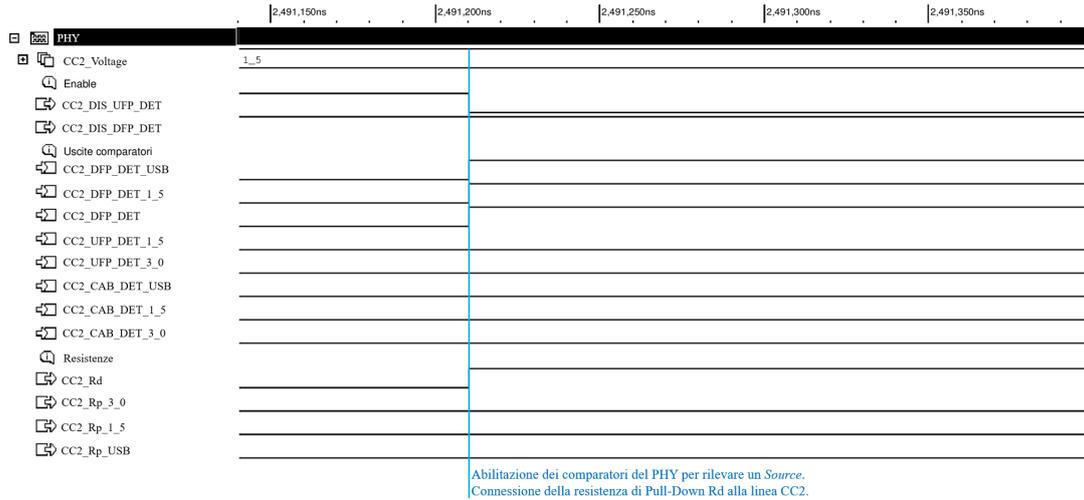


Figura 5.16: Input e Output PHY dopo aver impostato l'IP come Sink ed aver abilitato la linea CC2

Come visibile dalla figura 5.17, `TypeCVstateCC2` assume valore pari a "11"<sup>2</sup>. Di conseguenza, il segnale `CC2_Level_11` viene attivato e nei cicli successivi `CC2_Level_11_q` e `CC2_Level_11_qq`, e il contatore `CC2_Counter` viene resettato a "110". La macchina a stati passa dallo stato di RESET allo stato UNSTABLE. Essendo che il livello di tensione sulla linea CC2 non cambia e non vengono generati degli eventi interni dovuti ai segnali `ANAMODE`, `ANASUBMODE` e `CCEnable`, il contatore decresce ogni volta che il segnale è attivo il segnale `HBitClkEn` durante il fronte di salita di `USBClk`. Quando il contatore arriva a "001", viene attivato il `CC2_TCounter` e il segnale `CC2_Level_11_Stable` viene posto a '1'. La macchina a stati passa quindi dallo stato UNSTABLE allo stato `STABLE_01`. Di conseguenza viene generato l'evento `CC2Evt`.

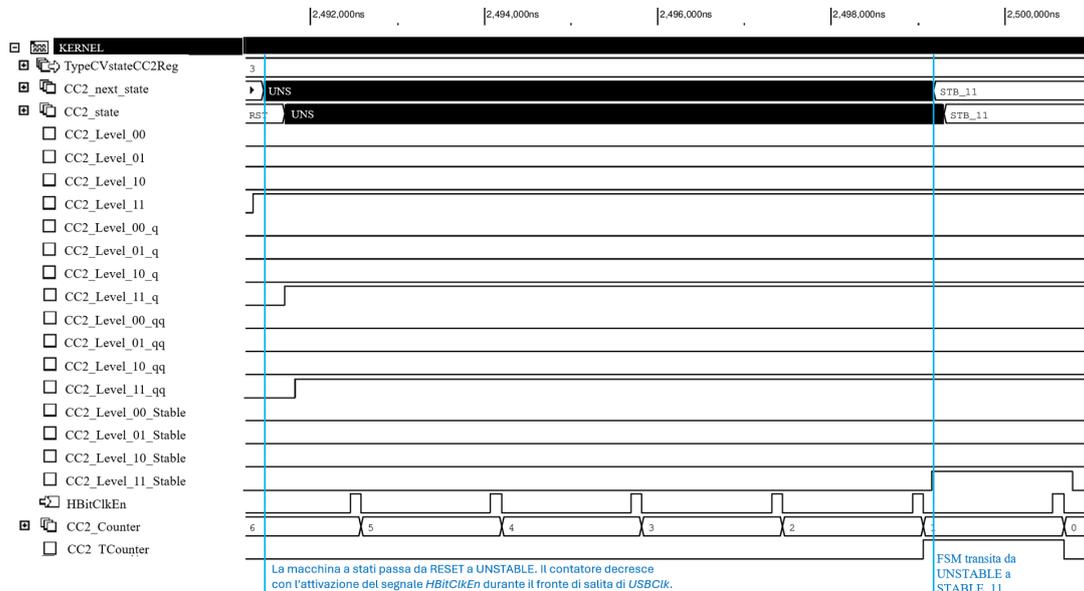


Figura 5.17: FSM con CC2 a 1.5V

<sup>2</sup>`ANAMODE` = '1', `CCEnable` = "10", `CC2_DFP_DET` = '1', `CC2_DFP_DET_USB` = '1', `CC2_DFP_DET_1_5` = '1'.

Infine l'evento TYPECEVT2 e il valore attuale di tensione nella linea CC2, TYPECVSTATECC2, vengono scritti all'interno dello SR (fig. 5.18).

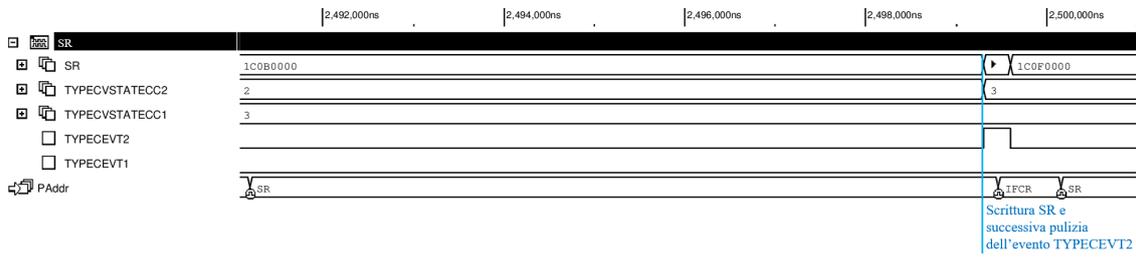


Figura 5.18: Scrittura TYPECEVT2 e TYPECVSTATECC2 in SR, CC2\_Voltage = 1.5V

## 5.2 Sintesi

La sintesi del controllore `Type-C only` è stata effettuata utilizzando il software `Synopsys Design Compiler`, impiegando il processo tecnologico `TSMC 90 nm`. Nella fase di sintesi, il design creato a livello `Register Transfer Level (RTL)` viene convertito in una `netlist` a livello di porte logiche, ottimizzata per area, timing e consumo energetico. Durante la sintesi, il tool ottimizza il design in base ai vincoli imposti, valutando l'area totale occupata dalle celle logiche e i tempi di propagazione dei segnali lungo i percorsi critici.

I clock utilizzati nel design sono il `PClk`, con una frequenza di 75 MHz, e il `USBClk`, con una frequenza di 18 MHz.

### 5.2.1 Report Area

Il report dell'area, riportato in figura 5.19, mostra l'occupazione totale del design in termini di celle logiche. L'area complessiva del progetto è espressa in micrometri quadrati ( $\mu\text{m}^2$ ) e si suddivide in area combinatoria e sequenziale. L'area combinatoria include le porte logiche, mentre l'area sequenziale è associata alle celle sequenziali, come flip-flop e latch. Nel report è anche indicata l'area equivalente, espressa in `kGates`, che rappresenta il numero di gate `NAND2` equivalenti utilizzati per rappresentare l'area totale del design. Un gate `NAND2` è una delle unità logiche più semplici e comuni in tecnologia `CMOS`, e serve come riferimento per confrontare l'area tra diversi design e tecnologie. Nel caso del design `Type-C only`, l'area complessiva ottenuta è di 2.75 `kGates`, indicando che, se il progetto fosse realizzato esclusivamente con porte `NAND2`, ci sarebbero 2750 di queste porte logiche.

```

*****
Software: Synthesis Tool Synopsys Design Compiler
Report : area
Design : typeC_only
Process: TSMC090 ( reference ND2 used: 2.19521 um2)
Clock Frequency: PClk = 75MHz ; USBClk = 18MHz
*****

Number of ports:                1256
Number of nets:                 1863
Number of cells:                1282
Number of combinational cells:   755
Number of sequential cells:      493
Number of macros/black boxes:    0
Number of buf/inv:              262
Number of references:           20

Combinational area:             2383.987164
Buf/Inv area:                   915.398386
Noncombinational area:          3650.617611
Macro/Black Box area:           0.000000

Total cell area:                 6034.604775
Equivalent area:                 2,75 kGates

```

Figura 5.19: Report area

## 5.2.2 Report Timing

Il report del timing fornisce informazioni sui percorsi critici del design, che determinano la frequenza operativa massima del sistema. Il percorso critico rappresenta il cammino più lento lungo il quale un segnale deve propagarsi all'interno del circuito, influenzando quindi la velocità massima con cui il design può funzionare senza introdurre violazioni di temporizzazione.

Di seguito sono riportati due esempi di percorsi critici associati ai due clock del sistema.

### Percorso critico associato a PClk

Il percorso critico associato al clock PClk, riportato in figura 5.20, inizia dal segnale di ingresso PEnable, sincronizzato con PClk, e termina sull'uscita PRData[15]. Il tempo di arrivo del dato lungo questo percorso è di 7.85 ns, mentre il tempo richiesto per una corretta propagazione è di 8.03 ns. Questa differenza produce uno slack positivo di 0.18 ns, il che significa che il percorso rispetta i vincoli temporali definiti per il clock a 75 MHz. Avere uno slack positivo è fondamentale perché indica che il segnale ha un margine di tempo sufficiente per propagarsi attraverso il circuito, riducendo il rischio di violazioni temporali.

```

Software: Synthesis Tool Synopsys Design Compiler
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : typeC_only
Process: TSMC090 ( reference ND2 used: 2.19521 um2)
Clock Frequency: PClk = 75MHz ; USBClk = 18MHz
*****

Operating Conditions: WCLCOM Library: tcbn90lpefbwp14thvtwcl_ccs
Wire Load Model Mode: segmented

Startpoint: PEnable (input port clocked by PClk)
Endpoint: PRData[15] (output port clocked by PClk)
Path Group: PClk
Path Type: max

Des/Clust/Port      Wire Load Model      Library
-----
typec_only          ZeroWireload         tcbn90lpefbwp14thvtwcl_ccs
typec_only_itf     ZeroWireload         tcbn90lpefbwp14thvtwcl_ccs

Point                                     Incr      Path
-----
clock PClk (rise edge)                   0.00      0.00
clock network delay (ideal)              0.00      0.00
input external delay                     2.78      2.78 r
PEnable (in)                             0.02      2.80 r
I295/Z (CKBD1BWP7THVT)                  0.15      2.95 r
i_itf/PEnable (typec_only_itf)          0.00      2.95 r
i_itf/I43/ZN (IND3D1BWP7TULL)           0.32      3.27 f
i_itf/I242/ZN (NR2XD0BWP7THVT)         0.29      3.56 r
i_itf/I72/ZN (INVD1BWP7THVT)           0.14      3.71 f
i_itf/I68/ZN (NR3D0BWP7TULL)           0.75      4.46 r
i_itf/I106/ZN (CKND0BWP7TULL)           0.68      5.13 f
i_itf/I212/ZN (OAI211D0BWP7TULL)       0.56      5.70 r
i_itf/PRData[15] (typec_only_itf)       0.00      5.70 r
I282/Z (BUFFD6BWP7TULL)                2.15      7.85 r
PRData[15] (out)                        0.00      7.85 r
data arrival time                        7.85

clock PClk (rise edge)                   13.89     13.89
clock network delay (ideal)              0.00     13.89
clock uncertainty                         -0.30     13.59
output external delay                    -5.56     8.03
data required time                       8.03

data required time                       8.03
data arrival time                        -7.85
-----
slack (MET)                              0.18

```

Figura 5.20: Report Critical Path PClk

### Percorso critico associato a USBClk

Il percorso critico associato al clock USBClk, riportato in figura 5.21, si sviluppa all'interno del modulo `Clock_Divider`, inizia dal registro `SClk_int_reg`, attivato sul fronte di salita del clock, e termina sul latch `IQ_reg`, sensibile al livello negativo. Il tempo di arrivo del dato lungo questo percorso è di 1.55 ns, mentre il tempo richiesto per una corretta propagazione è di 27.48 ns. Questa differenza produce uno slack positivo di 25.93 ns, il che significa che il percorso rispetta ampiamente i vincoli temporali definiti per il clock a 18 MHz.

```

Startpoint: i_clock_divider/SClk_int_reg
              (rising edge-triggered flip-flop clocked by USBClk)
Endpoint: i_clock_divider/i_clktree_cgating_sclk/IQ_reg
              (negative level-sensitive latch clocked by USBClk)
Path Group: USBClk
Path Type: max

Des/Clust/Port          Wire Load Model          Library
-----
st_notech_ctcghls      ZeroWireload             tcbn90lpefbwp14thvtwcl_ccs
st_notech_rst_syncs0   ZeroWireload             tcbn90lpefbwp14thvtwcl_ccs
typeC_only             ZeroWireload             tcbn90lpefbwp14thvtwcl_ccs
typeC_only_clock_divider ZeroWireload             tcbn90lpefbwp14thvtwcl_ccs

Point                    Incr          Path
-----
clock USBClk (rise edge)          0.00         0.00
clock network delay (ideal)       0.00         0.00
i_clock_divider/SClk_int_reg/CP (SDFCNQD0BWP7TULL)  0.00         0.00 r
i_clock_divider/SClk_int_reg/Q (SDFCNQD0BWP7TULL)  1.06         1.06 r
i_clock_divider/i_clktree_cgating_sclk/E (st_notech_ctcghls)  0.00         1.06 r
i_clock_divider/i_clktree_cgating_sclk/I3/Z (OR2D0BWP7TULL)  0.49         1.55 r
i_clock_divider/i_clktree_cgating_sclk/IQ_reg/D (LNQD1BWP7TULL)  0.00         1.55 r
data arrival time                1.55         1.55

clock USBClk (fall edge)          27.77        27.77
clock network delay (ideal)       0.00         27.77
clock uncertainty                  -0.30        27.48
i_clock_divider/i_clktree_cgating_sclk/IQ_reg/EN (LNQD1BWP7TULL)  0.00         27.48 f
time borrowed from endpoint       0.00         27.48
data required time                27.48        27.48

data required time                27.48
data arrival time                 -1.55

slack (MET)                       25.93

```

Figura 5.21: Report Critical Path USBClk

## 5.3 Conclusioni

Le simulazioni effettuate confermano il corretto funzionamento del sistema, evidenziando l'affidabilità della soluzione sviluppata. Inoltre, la sintesi mostra il rispetto dei requisiti di timing, dimostrando che il design è in grado di operare in modo efficace. Uno degli aspetti più significativi è l'ottimizzazione ottenuta in termini di area: la soluzione sviluppata consente, per **STMicroelectronics**, un risparmio di circa il 78% rispetto alle attuali soluzioni. Questo risultato sottolinea non solo l'efficienza della soluzione proposta, ma anche il suo potenziale per ridurre i costi di produzione dei dispositivi che implementano il protocollo **USB Type-C**.

# Ringraziamenti

Desidero esprimere la mia profonda gratitudine a tutte le persone che hanno contribuito a rendere possibile questo traguardo.

In primo luogo, ringrazio il mio relatore, il *Professore Danilo Demarchi*, per avermi guidato con dedizione lungo questo percorso.

Un grazie di cuore all'*Ingegnere Giuseppe Guarnaccia*, all'*Ingegnere Giovanni Pangallo* e a tutto il team Digital IP di STMicroelectronics Catania per avermi accolto calorosamente, rendendo la mia esperienza indimenticabile.

Al mio collega e amico *Dave*, va il mio grazie infinito per le notti insonni trascorse insieme a scrivere codice VHDL. Per ogni progetto realizzato, per gli occhi rossi e per quei momenti in cui pensavamo di non farcela, ma alla fine ce l'abbiamo fatta. Ricorderò sempre la nostra gioia quando abbiamo visto finalmente funzionare il RISC-V.

Un sentito ringraziamento al *Professore Fiorino*, che è stato un mentore prezioso e una guida costante, dal liceo fino a oggi.

A *Sonia*, la mia costante in un mare di variabili: il tuo amore è stata la mia forza durante questo viaggio. Grazie per esserci sempre e per rendere la mia vita straordinaria. Ti amo infinitamente.

Un ringraziamento di cuore va a mia *Madre*, per avermi insegnato a trovare la forza anche nei momenti difficili, e a mio *Padre*, per avermi trasmesso l'importanza del sacrificio e della dedizione. Il vostro infinito amore e sostegno hanno reso possibile ogni mio traguardo. Ogni successo che ho raggiunto è anche il vostro successo. Senza di voi, nulla di tutto questo sarebbe stato possibile. Vi sarò per sempre grato per tutto ciò che fate per me.

Un abbraccio speciale a mia sorella *Sabrina*. Grazie per essere sempre stata al mio fianco, la tua presenza è un dono prezioso. Sei la sorella migliore che potessi desiderare e ti voglio un mondo di bene.

Ai miei *Nonni*, *Zii* e *Cugini*, un sentito ringraziamento per essere sempre fieri di me. Il vostro sostegno mi ha spinto a perseguire i miei obiettivi.

Un grazie particolare va agli *Amici di sempre*, che sono rimasti al mio fianco nonostante la distanza. Grazie per i momenti di gioia condivisi e per la leggerezza che avete portato nella mia vita.

Un ringraziamento speciale va a tutti i *Nuovi amici* conosciuti a Torino. Grazie per aver trasformato questa città in una seconda casa. La vostra amicizia ha arricchito la mia esperienza universitaria in modi inaspettati e preziosi.

Desidero esprimere un ringraziamento particolare a *Nunzio*, il mio confidente sempre pronto ad ascoltarmi, e a *Carmelo*, il mio compare di viaggi. Spero che insieme riusciremo a realizzare i nostri sogni e a dare vita alle tante aziende di cui fantastichiamo.

Infine, voglio ricordare chi è stato parte del mio inizio e purtroppo non c'è più: mio *Nonno Calogero*, mio *Zio Lillo* e mio *Zio Stefano*. La vostra memoria è sempre viva nel mio cuore.

# Bibliografia

- [1] USB-IF. *About USB-IF*. URL: <https://usb.org/about>.
- [2] Wikipedia contributors. *USB hardware – Connector types*. URL: [https://en.wikipedia.org/w/index.php?title=USB\\_hardware&oldid=1227524215](https://en.wikipedia.org/w/index.php?title=USB_hardware&oldid=1227524215).
- [3] USB-IF. *Universal Serial Bus Type-C Cable and Connector Specification*. URL: <https://www.usb.org/sites/default/files/USB%20Type-C%20Spec%20R2.0%20-%20August%202019.pdf>.
- [4] STMicroelectronics. *Introduction to USB Type-C Power Delivery for STM32 MCUs and MPUs*. URL: [https://www.st.com/resource/en/application\\_note/dm00536349-usb-type-c-power-delivery-using-stm32xx-series-mcus-and-stm32xxx-series-mpus-stmicroelectronics.pdf5](https://www.st.com/resource/en/application_note/dm00536349-usb-type-c-power-delivery-using-stm32xx-series-mcus-and-stm32xxx-series-mpus-stmicroelectronics.pdf5).
- [5] STMicroelectronics. *RM0444 Reference manual – USB Type-C / USB Power Delivery interface*. URL: [https://www.st.com/resource/en/reference\\_manual/rm0444-stm32g0x1-advanced-armed-32bit-mcus-stmicroelectronics.pdf](https://www.st.com/resource/en/reference_manual/rm0444-stm32g0x1-advanced-armed-32bit-mcus-stmicroelectronics.pdf).