

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

FACOLTA' DI INGEGNERIA

**CORSO DI LAUREA SPECIALISTICA IN INGEGNERIA
ELETTRONICA**

**Progetto di un'unità di elaborazione per una telecamera
stereo basata su FPGA Xilinx**

CANDIDATO:
Francesco Betti

RELATORE:
Ing. Stefano Mattoccia

CORRELATORI:
Prof. Giovanni Neri
Ing. Michele Borgatti
Ing. Davide Nanni

Anno Accademico 2010/2011

Sessione II

Indice

1	Visione stereoscopica e telecamera stereo 3D	1
1.1	Introduzione	1
1.2	Il lavoro svolto	4
2	Studio e progettazione	7
2.1	Problematiche, specifiche e datapath	7
2.2	FPGA	10
2.3	Memorie	15
2.3.1	Flash PROM	15
2.3.2	DDR	18
2.3.3	FIFO	19
2.4	Periferiche	21
2.4.1	Clock	21
2.4.2	Reset	24
2.4.3	Protezione	25
2.5	Alimentazione	26
3	Realizzazione degli schematici	33
3.1	Schema FPGA	34
3.2	Schema DDR	37
3.3	Schema FIFO	38
3.4	Schema FIFO e DDR: alimentazione	38
3.5	Schema PROM	38
3.6	Schema periferiche	39
3.7	Connettori	39
3.8	Alimentazioni generali	40
3.9	Note e disposizione	41

3.10 Schematico	41
4 Conclusioni	45
4.1 Risultati	45
4.2 Sviluppi futuri	46
Appendice	46
A Schematici	47
B File UCF	65
C Layout	73
Bibliografia	88
Ringraziamenti	93

Elenco delle figure

1.1	Proiezione di due punti sullo stesso punto del piano immagine	2
1.2	Principio di funzionamento di un sistema stereoscopico	3
1.3	Piano e vincolo epipolare	4
2.1	Ipotetico schema a blocchi della scheda da realizzare	9
2.2	Banchi di I/O per tutti i dispositivi LX4, LX9, LX16, LX25 e LX45 e per gli LX75, LX100 e LX150 nei package CSG484 e FG(G)484	10
2.3	Pin disponibili/Dispositivo/Package	11
2.4	Numero di pin di I/O per banco per ogni dispositivo	12
2.5	Disposizione dei pin dei dispositivi LX45 e LX75	13
2.6	SSO per coppia di Vcco/GND	14
2.7	Schema a blocchi delle memorie Flash PROM	16
2.8	Setup di configurazione in modalità Daisy-Chain con FPGA Master e Slave	17
2.9	Schema a blocchi della memoria DDR	19
2.10	Schema a blocchi della memoria FIFO	20
2.11	Connessione BUFGMUX nel banco 0 e 1	21
2.12	Risorse del clock globale condivise tra banchi 0 e 1	22
2.13	Schema a blocchi del circuito di generazione del Clock	24
2.14	Applicazione tipica del Max811/812	25
2.15	Principio di funzionamento del sistema di protezione	26
2.16	Consumi massimi	27
2.17	Schema interno del convertitore switching	28
2.18	Schema interno del convertitore LDO ADP1706	29
2.19	Schema interno del DDR SINK/SOURCE	31
3.1	Modalità di configurazione delle Spartan 6	36

3.2	Schema a blocchi della scheda da realizzare	43
A.1	Alimentazione switching e sink/source ddr	48
A.2	Alimentazione LDO e Reset	49
A.3	Filtro di alimentazione FPGA	50
A.4	FPGA: banco 0	51
A.5	FPGA: banco 1	52
A.6	FPGA: banco 2	53
A.7	FPGA: banco 3	54
A.8	FPGA: banco alimentazioni e segnali dedicati	55
A.9	Memoria DDR3	56
A.10	Memorie Flash PROM	57
A.11	Memoria FIFO	58
A.12	Alimentazione FIFO e DDR3	59
A.13	Circuiti elettronici periferici	60
A.14	Connettori di ingresso	61
A.15	Connettori di uscita	62
A.16	Note informative	63
A.17	Disposizione dei dispositivi	64
C.1	Lato componenti	74
C.2	Piano interno di massa	75
C.3	Piano interno alimentazioni	76
C.4	Piano interno alimentazioni e segnali	77
C.5	Piano interno massa	78
C.6	Piano interno segnali	79
C.7	Piano interno massa	80
C.8	Lato saldature	81
C.9	Solder lato componenti	82
C.10	Solder lato saldature	83
C.11	Pasta salda SMD lato componenti	84
C.12	Pasta salda SMD lato saldature	85
C.13	Specifiche di foratura	86
C.14	Serigrafia lato componenti (solo componenti through-hole)	87

Capitolo 1

Visione stereoscopica e telecamera stereo 3D

1.1 Introduzione

La visione stereoscopica ha riscosso un notevole interesse in ambito scientifico ed industriale poiché è in grado di fornire immagini tridimensionali utilizzando telecamere tradizionali e non imponendo alcun vincolo sull'oggetto ripreso, che può quindi essere libero di muoversi. La visione stereoscopica vuole riprodurre il funzionamento degli occhi umani che registrano immagini da due punti di vista differenti; le immagini vengono inviate al cervello che incoscientemente le elabora e ci fornisce un'immagine tridimensionale. I primi studi moderni furono condotti dal fisico inglese Sir Charles Wheatstone (Glocester 1802 - Parigi 1875) che nel 1832 costruì uno stereoscopio con lenti e specchi, il quale consentiva di vedere un'immagine tridimensionale a partire da due immagini affiancate. Il principio alla base della visione stereoscopica, noto sin dal rinascimento, consiste in una triangolazione mirata a mettere in relazione la proiezione di un punto della scena sui due (o più) piani immagine che compongono il sistema di visione stereoscopico. L'individuazione dei punti omologhi, problema noto in letteratura come il problema della corrispondenza (correspondence problem o matching stereo), consente di ottenere una grandezza denominata disparità (disparity) mediante la quale, conoscendo opportuni parametri del sistema stereoscopico, è possibile risalire alla posizione 3D del punto considerato. Il sistema generico di partenza per capire come funziona è quello mostrato in figura 1.1 in cui si ha una telecamera

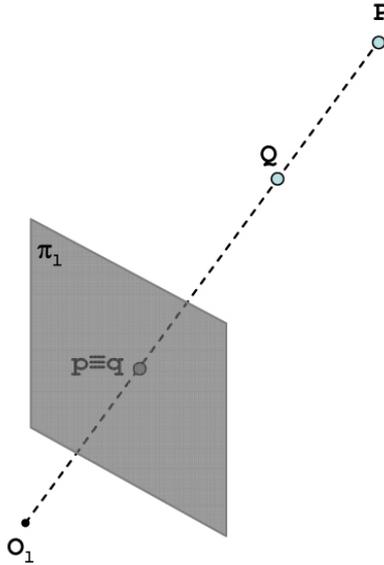


Figura 1.1: Proiezione di due punti sullo stesso punto del piano immagine

e vi sono due punti distinti giacenti sulla retta che passa dal centro ottico della telecamera e che si trovano nello stesso punto nel piano immagine. Inserendo nel sistema una seconda telecamera, tra tutti i punti che sono sulla retta precedentemente menzionata, al più un solo punto (per esempio q'), che sarà chiamato “*punto omologo*”, verrà proiettato sul piano immagine della seconda telecamera. La determinazione dei punti omologhi consente di risalire alle coordinate dei punti nello spazio, mediante una procedura denominata “*triangolazione*”. La ricerca dei punti omologhi, chiamato “*problema delle corrispondenze*”, richiederebbe una ricerca su un piano bidimensionale. Sfruttando però una caratteristica del sistema stereoscopico è possibile limitare la ricerca ad uno spazio monodimensionale, una retta (“*vincolo epipolare*”), che è il risultato dell’intersezione tra il “*piano epipolare*”, definito dai due centri ottici e la retta passante per il primo centro ottico e i punti P e Q, e il piano immagine della seconda telecamera.

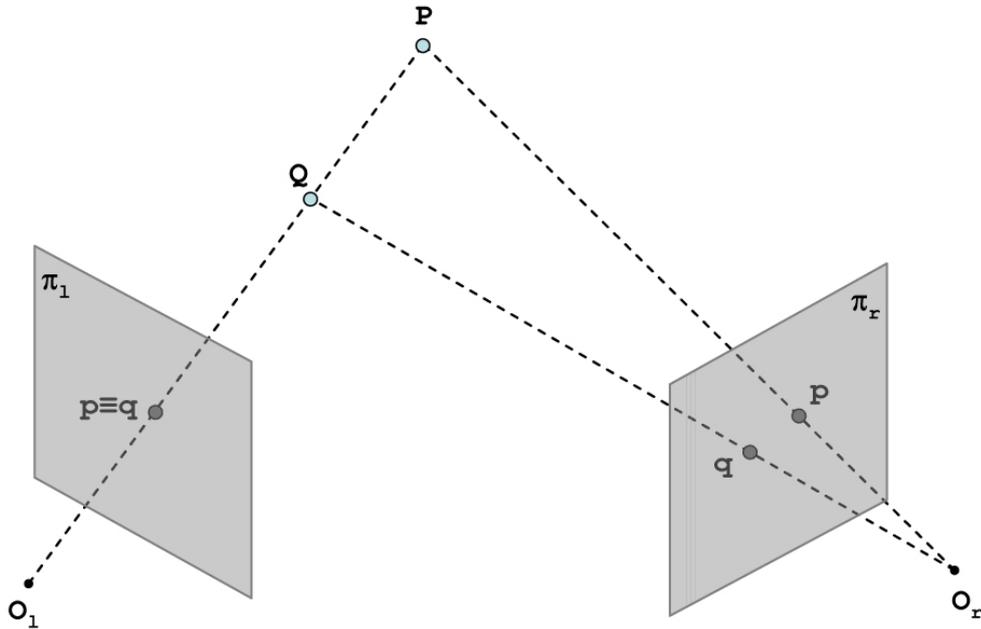


Figura 1.2: Principio di funzionamento di un sistema stereoscopico

Questo approccio consente di semplificare notevolmente la complessità algoritmica; spetterà ad un calcolatore trovare i punti omologhi e quindi avere le coordinate tridimensionali dei vari punti dello spazio. Il mio lavoro si colloca all'interno di un progetto di ricerca che mira a realizzare una telecamera stereo 3D con unità di calcolo a bordo su FPGA: su di esso verranno mappati gli algoritmi necessari all'elaborazione delle immagini. Le telecamere sono due sensori in tecnologia cmos, ciascuna delle quali cattura un'immagine alla risoluzione di 640x480 pixel con un frame rate di 60 fps; essendo le telecamere sincronizzate cattureranno contemporaneamente le immagini e invieranno nello stesso momento i dati da processare. Sull'FPGA verranno implementati gli algoritmi che consentono di ottenere una mappa 3D dell'immagine la quale viene inviata ad un Host (PC) esterno in modo da poter memorizzare i dati e renderli leggibili.

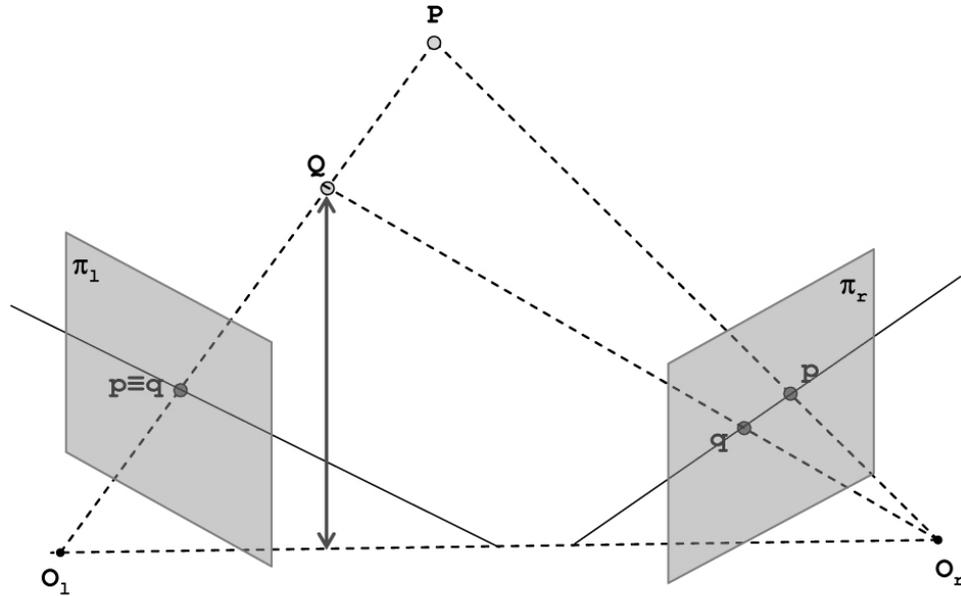


Figura 1.3: Piano e vincolo epipolare

L'oggetto di studio della presente tesi è la realizzazione di una scheda elettronica con FPGA per l'elaborazione dei dati provenienti da due telecamere.

1.2 Il lavoro svolto

Il lavoro svolto è cominciato ponendo alcuni vincoli nelle specifiche del sistema, restrizioni che hanno portato a effettuare alcune scelte progettuali. Nel prossimo capitolo innanzitutto si parlerà delle problematiche riscontrate durante la progettazione e delle scelte effettuate, fatte al fine di realizzare una scheda compatibile con quanto già si possiede, ma anche in grado di aggirare i problemi che si sono verificati fin'ora; verrà poi mostrato uno schema a blocchi della scheda. Successivamente si presenterà poi lo studio dei vari componenti a partire dall'FPGA per poi passare alle varie memorie necessarie oppure non indispensabili per il corretto funzionamento della scheda. In seguito verranno analizzati i circuiti periferici e i dispositivi di alimentazione per poi terminare

con una breve panoramica dei connettori e degli elementi aggiuntivi per avere una maggiore flessibilità. Nel capitolo successivo si parlerà di come è stato realizzato lo schematico in seguito allo studio preliminare, mentre nell'ultimo capitolo saranno presenti le conclusioni e gli sviluppi futuri di questo progetto. Le appendici contengono rispettivamente gli schemi elettrici Orcad, il file UCF necessario alla programmazione della FPGA ed infine il piazzamento e il layout finale della scheda.

Capitolo 2

Studio e progettazione

2.1 Problematiche, specifiche e datapath

Il sistema di partenza prevedeva una scheda Spartan 6 sp605, una scheda con deserializzatore, una scheda con controller usb, una scheda di alimentazione e due schede per i sensori (un master e uno slave). Il sensore slave è connesso al sensore master: a sua volta, trasferisce i dati tramite una porta seriale alla scheda deserializzatore sulla quale si trovano un deserializzatore e un microcontroller per la programmazione; questa scheda ha il compito di inviare all'unità di elaborazione i dati in forma parallela. Dopo aver elaborato i segnali con gli opportuni algoritmi, i risultati vengono trasferiti ad un Host tramite un scheda avente un controller USB 2.0. I problemi riscontrati nell'attuale sistema riguardano per lo più i cavi che introducono notevoli disturbi alle alte frequenze e la connessione con la scheda USB. Il controller Usb infatti richiede una memoria FIFO che inizialmente era stata mappata sull'FPGA, soluzione scomoda in quanto richiede risorse, inoltre l'USB 2.0 può trasferire al massimo ad una velocità di 60MB/s. Questa velocità è quella di punta e generalmente non viene raggiunta, ma considerando di dover trasferire all'host i dati provenienti dalle due videocamere, che sono ad una risoluzione di 640x480 con una profondità di colore a 16bit e 60fps, si ottiene un trasferimento di circa 74MB/s che è superiore a quello disponibile con l'USB 2.0. La scheda dovrà avere la possibilità di collegarsi anche a una scheda non ancora realizzata capace di sostenere una tale velocità di trasferimento di dati ed il sistema dovrà prevedere anche un connettore ausiliario con uscita LVDS dei dati. La scheda del deserializzatore, sarebbe opportuno provare a bypassarla in modo da ridur-

re i connettori mentre si dovrà integrare la parte di alimentazione (ingresso a 12V). Inizialmente verrà utilizzato un FPGA LX45 identico a quello montato sulla SP605: le prestazioni saranno migliori rispetto a quelle che si hanno utilizzando l'evaluation board, ma non si esclude a priori di poter montare in futuro un FPGA con maggiori prestazioni. Si richiede quindi di effettuare uno studio sulla compatibilità tra i vari integrati al fine di poter utilizzare diversi componenti sulla stessa scheda ed eventuale circuiteria esterna per avere la compatibilità, senza dimenticare di cercare di minimizzare gli spazi occupati; si dovrà integrare inoltre sulla scheda anche una memoria DDR che possa essere montata o meno a seconda delle esigenze. Infine, per quanto concerne l'FPGA, si dovranno cercare soluzioni che consentano di proteggere il software. Altri dettagli sulla scheda riguardano l'aggiunta di 3 led e 3 ingressi comandabili con jumper per poter effettuare dei test. Un primo datapath è quello mostrato nella figura seguente in cui compaiono le componenti principali del sistema, ma non è stato ancora definito come verranno connessi tali componenti durante il successivo sviluppo del prototipo. Le specifiche di questa scheda dovranno essere:

- FPGA LX45
- Memoria Flash PROM
- Memoria DDR
- Memoria FIFO
- Protezione dati di configurazione
- Connettore Hirose 100 pin per deserializzatore
- Connettore 40 pin per scheda USB 2.0
- Connettore per una nuova scheda
- Connettori ausiliari per dati e alimentazioni
- Circuiteria periferica (Alimentazione, Clock, Reset, Led...)

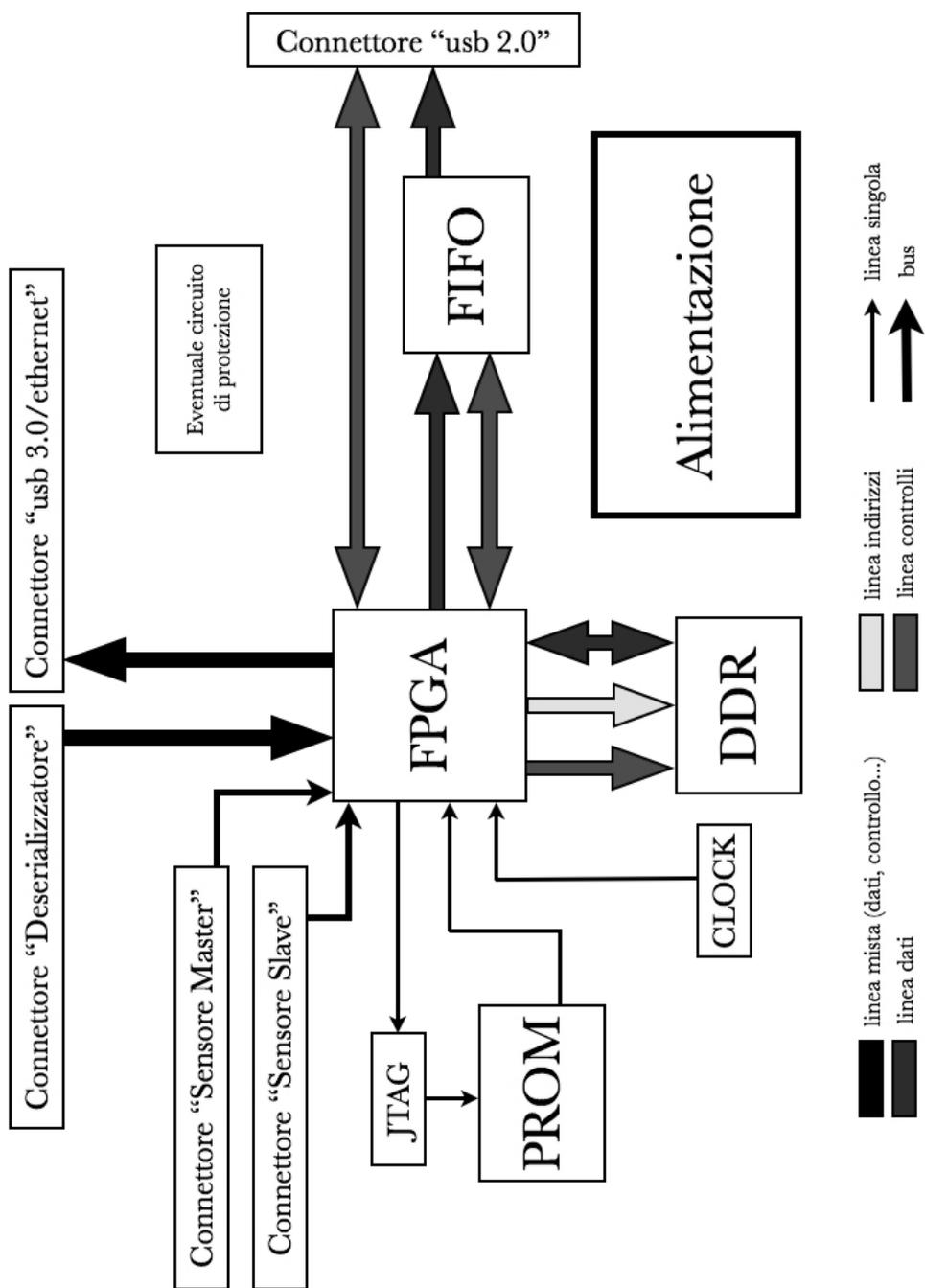


Figura 2.1: Ipotetico schema a blocchi della scheda da realizzare

2.2 FPGA

Lo studio sulle FPGA da utilizzare nel progetto è cominciato cercando le informazioni sulla struttura interna e sulla compatibilità tra le differenti versioni di FPGA; è emerso che le FPGA internamente sono suddivise in banchi, per quanto riguarda i pin di I/O, che variano in numero a seconda del package utilizzato: per esempio le Spartan 6 aventi il package FG(G)484 hanno 4 banchi, uno per ogni lato del dispositivo, come mostrato in figura 2.2.

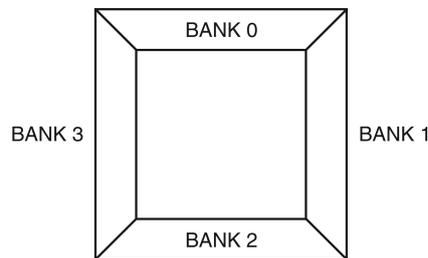


Figura 2.2: Banchi di I/O per tutti i dispositivi LX4, LX9, LX16, LX25 e LX45 e per gli LX75, LX100 e LX150 nei package CSG484 e FG(G)484

Sul file UG385 [7] è presente la risposta al problema di compatibilità: tutte le Spartan 6 LX con lo stesso package sono compatibili tra loro e tutte le Spartan 6 LXT con lo stesso package sono compatibili tra loro, tuttavia le Spartan6 LX non sono compatibili con le LXT pertanto viene scartata la possibilità di avere un transceiver sulla FPGA. Una nota riguarda i pin di I/O: i pin che non sono disponibili su un dispositivo vengono dichiarati come "Non Connessi" e questo preannuncia che si dovranno utilizzare solo i pin di I/O che tutti i dispositivi hanno in comune. La prima cosa da notare nella figura 2.3 è che gli unici tre package che possono essere utili sono FG(G)484, CSG484 e FG(G)676 in quanto FG(G)900 è disponibile solo per il dispositivo LX150 mentre il CSG324 non prevede FPGA con prestazioni superiori alla LX45. La scelta del package è ricaduta sul FG(G)484 per i seguenti motivi: il package FG(G)676 rende disponibili un numero di pin maggiore rispetto agli altri due, però è il package più ingombrante e i pin disponibili presenti con gli altri due package sono più che sufficienti al fine del progetto. Tra gli altri due package

è stato scartato il CSG484 poiché se in dimensioni è leggermente più piccolo (4mm per ogni lato) rispetto alla FG(G)484 questo tipo di package possiede un array di sfere più compatto e questo potrebbe causare problemi di allineamento e di saldatura in fase di assemblaggio.

Spartan-6 Device	User I/O Pins	Spartan-6 FPGA Package								
		TQG144	CPG196	CSG225	FT(G)256	CSG324	FG(G)484	CSG484	FG(G)676	FG(G)900
XC6SLX4	Available User I/Os	102	106	132	-	-	-	-	-	-
	Differential Pairs	51	53	66	-	-	-	-	-	-
XC6SLX9	Available User I/Os	102	106	160	186	200	-	-	-	-
	Differential Pairs	51	53	80	93	100	-	-	-	-
XC6SLX16	Available User I/Os	-	106	160	186	232	-	-	-	-
	Differential Pairs	-	53	80	93	116	-	-	-	-
XC6SLX25	Available User I/Os	-	-	-	186	226	266	-	-	-
	Differential Pairs	-	-	-	93	113	133	-	-	-
XC6SLX45	Available User I/Os	-	-	-	-	218	316	320	358	-
	Differential Pairs	-	-	-	-	109	158	160	179	-
XC6SLX75	Available User I/Os	-	-	-	-	-	280	328	408	-
	Differential Pairs	-	-	-	-	-	140	164	204	-
XC6SLX100	Available User I/Os	-	-	-	-	-	326	338	480	-
	Differential Pairs	-	-	-	-	-	163	169	240	-
XC6SLX150	Available User I/Os	-	-	-	-	-	338	338	498	576
	Differential Pairs	-	-	-	-	-	169	169	249	288
XC6SLX25T	Available User I/Os	-	-	-	-	190	250	-	-	-
	Differential Pairs	-	-	-	-	95	125	-	-	-
XC6SLX45T	Available User I/Os	-	-	-	-	190	296	296	-	-
	Differential Pairs	-	-	-	-	95	148	148	-	-
XC6SLX75T	Available User I/Os	-	-	-	-	-	268	292	348	-
	Differential Pairs	-	-	-	-	-	134	146	174	-
XC6SLX100T	Available User I/Os	-	-	-	-	-	296	296	376	498
	Differential Pairs	-	-	-	-	-	148	148	188	249
XC6SLX150T	Available User I/Os	-	-	-	-	-	296	296	396	540
	Differential Pairs	-	-	-	-	-	148	148	198	270

Figura 2.3: Pin disponibili/Dispositivo/Package

Scelto il package lo studio si è orientato sui pin di I/O: ogni banco possiede un numero di pin differente rispetto ad un altro banco dello stesso dispositivo e il numero di I/O di un dato banco è differente ovviamente dal numero di pin dello stesso banco di un altro dispositivo; in particolare nella figura 2.4 viene mostrato il numero di pin per ogni banco per ogni dispositivo.

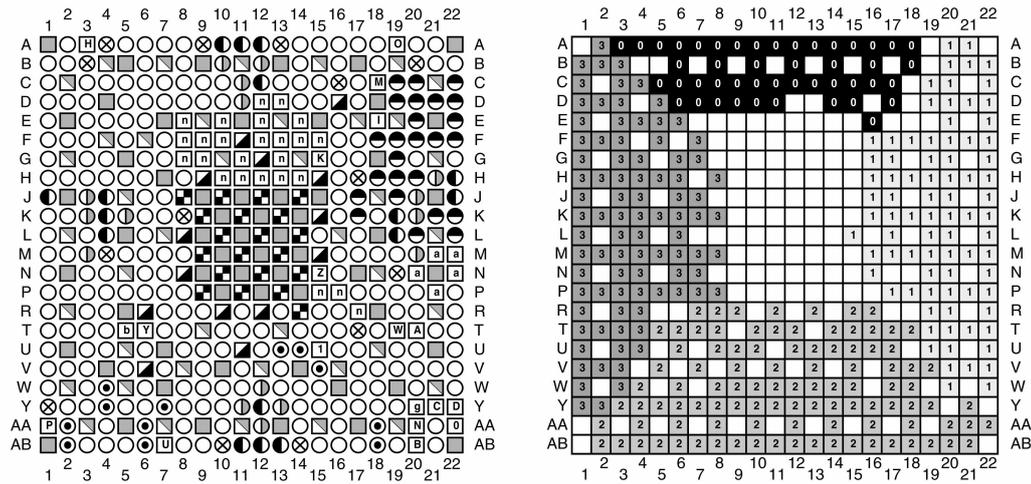
Device	Bank 0	Bank 1	Bank 2	Bank 3	Bank 4	Bank 5	Total I/O
LX45	46	82	100	88	N/A	N/A	316
LX75	56	82	54	88	N/A	N/A	280
LX100	68	82	88	88	N/A	N/A	326
LX150	68	82	100	88	N/A	N/A	338

Figura 2.4: Numero di pin di I/O per banco per ogni dispositivo

Si pu/’o notare un dato interessante: il trend del numero di I/O, tende a cresce all’aumentare della potenza del dispositivo tranne nel caso della LX75 che, contrariamente a quanto ci si aspetta, ha un numero di pin inferiore alla LX45 (280 contro i 316); inoltre si nota subito che se nei banchi 1 e 3 il numero di I/O è comune a tutti i dispositivi, nei banchi 0 e 2 il numero è differente pertanto saranno questi che dovranno essere tenuti sotto controllo in fase di definizione degli I/O per non avere delle incompatibilità. Proseguendo lo studio, si scopre che il dispositivo LX45 possiede un grande numero di pin NC nel banco 0 mentre la LX75 possiede un grande numero di pin inutilizzati nel banco 2 (la LX100 ha qualche pin NC nel banco 2 che si sovrappongono a quelli della LX75). Nella figura 2.5 vengono mostrati i pin, disponibili e non, delle due versioni e i banchi a cui sono associati. In una fase successiva si è focalizzato lo studio sui singoli banchi e sui singoli ingressi e uscite in quanto ad alte frequenze spesso si richiedono terminazioni di vario tipo. Gli ingressi differenziali possiedono all’interno una loro resistenza di 100ohm che elimina la necessità di metterne una esterna, gli ingressi single-ended dispongono di resistenze programmabili (25, 50 e 75 ohm) organizzate in una struttura pull-up e pull-down in modo da portare il livello di uscita a $V_{cc}/2$; anche l’impedenza di uscita è programmabile in modo da eliminare l’utilizzo di resistori esterni.

Ogni I/O è programmabile in modo da poter gestire differenti I/O standard come ad esempio LVCMOS, I2C e LVDS.

FG(G)484 Package—LX45



FG(G)484 Package—LX75

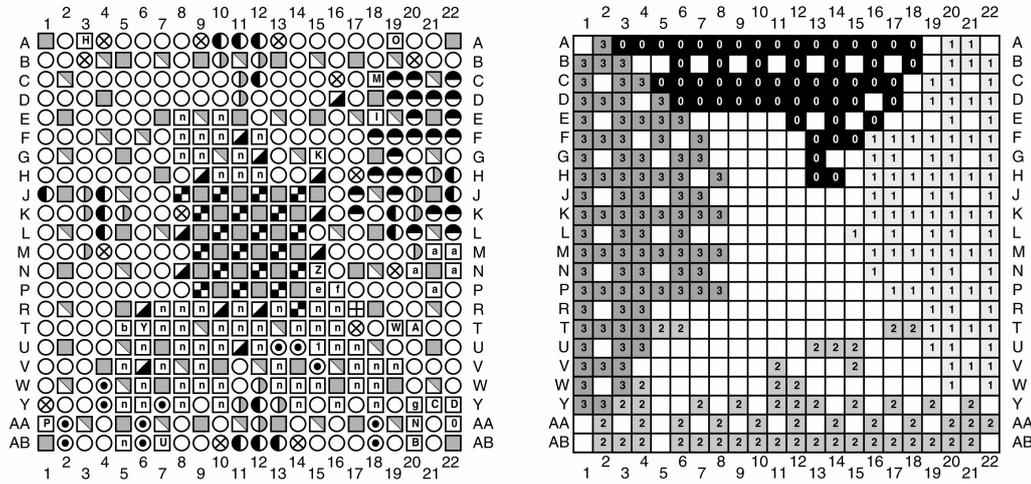


Figura 2.5: Disposizione dei pin dei dispositivi LX45 e LX75

In particolare si dovrà fare attenzione agli ultimi due standard citati in quanto l'I2C è uno standard open drain e pertanto richiede delle resistenze di pull-up, mentre lo standard LVDS sull'FPGA è disponibile in ingresso sugli I/O di ogni banco, mentre le uscite LVDS sono presenti solo sui banchi 0 e 2. Lo studio sugli ingressi rivela anche che vi sono delle informazioni riguardanti il numero di ingressi che possono commutare simultaneamente, infatti viene raccomandato un SSO ovvero Simultaneously Switching Outputs in relazione al numero di V_{cco}/GND e alla loro corrente di uscita: ogni I/O infatti può essere programmato per erogare 2, 4, 6, 8, 12, 16 o 24 mA di corrente ed ovviamente maggiori sono le uscite che commutano simultaneamente, maggiore sarà la corrente richiesta con il conseguente rischio di introdurre disturbi sull'alimentazione. Nella figura 2.6 sono riportati alcuni dati relativi la SSO dei vari banchi nel caso di uscite LVCMOS per ogni alimentazione.

V _{cco}	I/O Standard	Drive	Slew	SSO Limit per V _{cco} /GND Pair			
				All TQG144, CPG196, CSG225, FT(G)256, and LX devices in CSG324		All CSG484, FG(G)484, FG(G)676, FG(G)900, and LXT devices in CSG324	
				Bank 0/2	Bank 1/3	Bank 0/2	Bank 1/3/4/5
3.3V	LVCMOS33	2	Fast	42	46	42	44
			Slow	50	55	50	49
			QuietIO	60	68	60	60
		4	Fast	21	27	21	25
			Slow	32	37	32	32
			QuietIO	39	42	39	37
		6	Fast	14	19	14	17
			Slow	19	25	19	22
			QuietIO	29	30	29	25
		8	Fast	11	15	11	14
			Slow	15	20	15	18
			QuietIO	25	24	25	20
		12	Fast	1	3	1	1
			Slow	2	5	2	2
			QuietIO	4	9	4	7
		16	Fast	1	2	1	1
			Slow	1	5	1	1
			QuietIO	3	10	3	8
		24	Fast	1	2	1	1
			Slow	2	5	2	1
			QuietIO	7	9	7	7

Figura 2.6: SSO per coppia di V_{cco}/GND

Questi dati saranno utili sia in fase di progettazione hardware che in fase di programmazione: nel primo caso, si dovranno distribuire i pin di uscita tra i vari banchi, in modo da massimizzare le correnti erogabili; durante la programmazione si cercherà di erogare più corrente su un'uscita piuttosto che un'altra per migliorare le prestazioni del sistema. In questo progetto, volendo la compatibilità tra LX45 e dispositivi superiori, si avranno scelte quasi obbligate in quanto su due banchi gli I/O disponibili sono numericamente limitati. Proseguendo con lo studio, si è appurato che ogni banco possiede tensioni di alimentazione (V_{cc0} e V_{ref}) separate dagli altri banchi con il chiaro vantaggio di poter collegare a determinati pin, dispositivi con tensioni di alimentazione differenti qual'ora fosse necessario. In merito alla possibilità di bypassare il deserializzatore e connettere direttamente le uscite delle telecamere alla FPGA, si è trovata una soluzione: ogni tile possiede la logica per istanziare un deserializzatore che effettua una conversione seriale-parallelo dei dati in ingresso. Terminata questa analisi si è deciso di alimentare, dove possibile, i banchi a 3.3V, tensione che permette di avere un maggiore SSO.

2.3 Memorie

In questo capitolo verranno prese in considerazione tutte le memorie che dovranno essere presenti sulla scheda a cominciare dalla memoria di programmazione dell'FPGA per poi passare alla DDR e alla FIFO.

2.3.1 Flash PROM

Come appena accennato la FPGA deve essere programmata ogni volta che il sistema viene acceso. I dati per la sua configurazione sono memorizzati su un dispositivo esterno all'FPGA, una memoria Flash PROM: su di essa vengono caricati tramite porta JTAG. Per essere sicuri che il sistema funzioni correttamente, si utilizzerà una delle memorie messe a disposizione dalla Xilinx: XCF_{xx}S e XCF_{xx}P [8] [14].

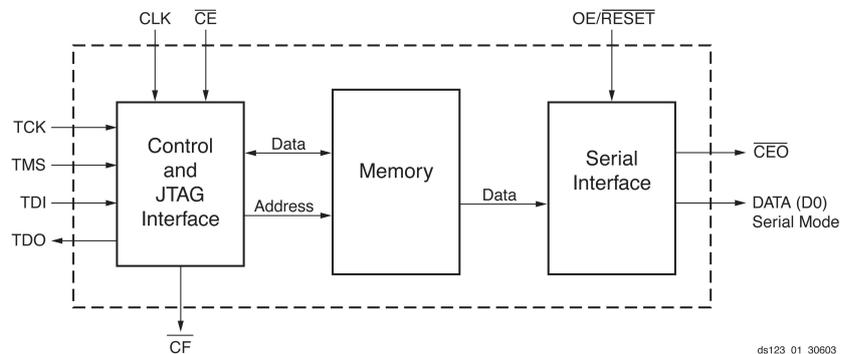


Figure 1: XCFxxS Platform Flash PROM Block Diagram

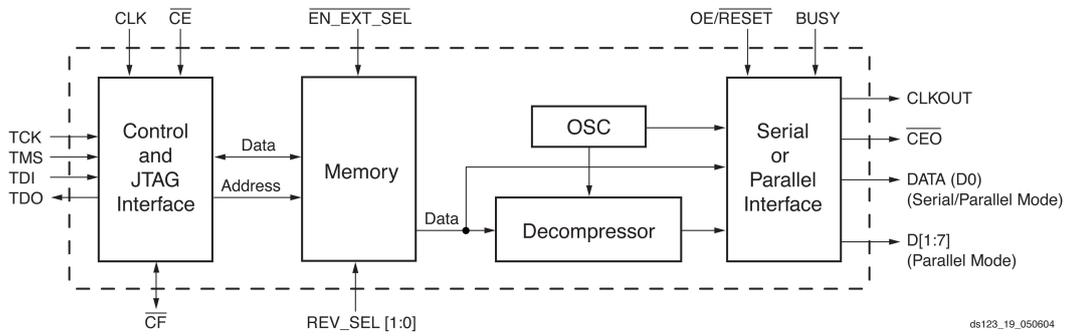


Figure 2: XCFxxP Platform Flash PROM Block Diagram

Figura 2.7: Schema a blocchi delle memorie Flash PROM

Sono memorie Flash di tipo NOR la prima delle quali possiede una capacità di memorizzazione di 1, 2 o 4 Mbit mentre la seconda può memorizzare 8, 16 o 32 Mbit, ed oltre ad essere più capiente può anche essere utilizzata per un trasferimento dati sia seriale che parallelo. Inoltre supportano il Design Revision, metodo della xilinx grazie al quale è possibile caricare direttamente più progetti su una singola memoria, e un sistema di compressione che permette di avere un file di configurazione più compatto e quindi spazio disponibile sulla memoria stessa che può essere utilizzato per il software. La programmazione avviene tramite porta JTAG ed i tool predisposti dalla Xilinx facilitano questa fase di progettazione. La FPGA LX45 necessita di un minimo di 16 Mbit per la sua configurazione pertanto verrà utilizzata la serie XCFxxP. Volendo rendere il sistema il più semplice possibile, si sceglierà una connessione seria-

le con una memoria tale da poter programmare sia la LX45 che i dispositivi più potenti come LX150. Quest'ultimo in particolare richiede due memorie: XCF32P+XCF08P. Siccome la configurazione non avviene tramite un dispositivo esterno, il metodo di configurazione dell'FPGA sarà di tipo Master Serial, in particolare essendo presenti due Prom la connessione sarà in Daisy Chain (figura 2.8).

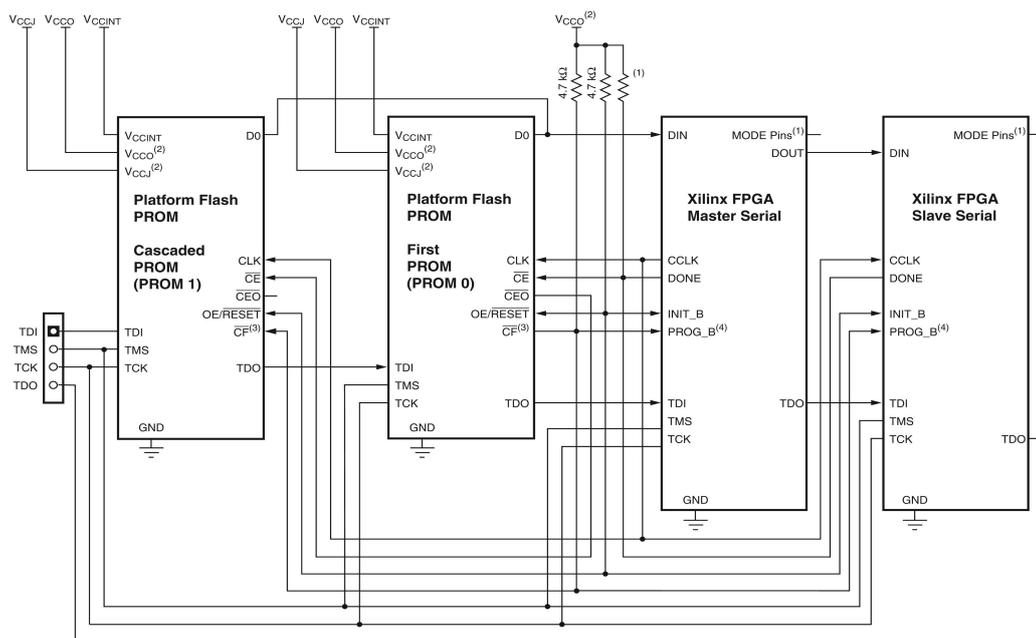


Figura 2.8: Setup di configurazione in modalità Daisy-Chain con FPGA Master e Slave

Questo sarà lo schema per la connessione delle PROM all'FPGA e ovviamente non ci sarà la seconda FPGA (Slave). Si pu/ò notare che c'è una resistenza senza valore in questo schema, ma è presente (330 ohm) in uno schema differente. Siccome 330 ohm con un'alimentazione di 3.3 V fornisce una corrente piuttosto elevata (10 mA) si è deciso di aumentare tale valore a 1 Kohm. Dal datasheet si vede che non vi sono integrati di dimensioni adeguate puramente seriali, mentre gli integrati, che possono svolgere la funzione di configurazione tramite trasferimento dati parallelo, possono essere utilizzati anche per il caricamento in modalità seriale. è importante capire come van-

no connessi i pin aggiuntivi presenti su questi circuiti integrati: in questo caso tutti i pin aggiuntivi si riferiscono alla modalità in parallelo o ad una modalità, Design Revision, che non è utilizzata e tali pin possono essere lasciati senza connessione in quanto inibiti da reti di pull-up e pull-down interne al dispositivo. Come già detto, la programmazione delle memorie avviene tramite porta JTAG. Si è evidenziato un problema: se si vuole riprogrammare la memoria, occorre spegnere il sistema e ricollegare la porta JTAG ad un Host. Può essere interessante invece caricare i nuovi dati sulla Flash durante il normale funzionamento: i dati passeranno attraverso la porta USB per poi essere scritti sulla PROM. Questo passaggio di dati può essere gestito da un componente sintetizzabile all'interno della FPGA chiamato picoBlaze che non è altro che un microcontroller. Per configurare la Prom il picoBlaze dovrà inviare i dati sulla linea JTAG in modo da inviarli alla memoria; questa operazione può essere utile anche per salvare sulla memoria Flash dati utili all'esecuzione del programma. Dovranno essere quindi utilizzati 4 pin di I/O da connettere ai segnali JTAG che possano essere però interrotti da jumper; in questo modo se qualcosa non dovesse funzionare la scheda risulterebbe comunque utilizzabile isolando questi 4 pin.

2.3.2 DDR

Come richiesto inizialmente, si è cercata una soluzione che prevedesse l'inserimento nel sistema di una memoria DDR possibilmente scollegabile. L'FPGA nel package FG(G)484 possiede al suo interno due MCB (Memory Controller Block) [10], oggetti embedded il cui scopo è interfacciarsi con una memoria DDR esterna all'FPGA. Gli MCB sono in grado di comunicare con memorie standard DDR, DDR2, DDR3, LPDDR e ognuno supporta un singolo componente a 4, 8 o 16 bit con una memoria fino a 4 Gb da cui si deduce che non è possibile montare per esempio una memoria standard da computer in quanto non possiede un singolo chip. Sono comunque consigliate alcune memorie sicuramente supportate dall'MCB di vario tipo soprattutto della Microchip. I Memory controller nella FPGA sono collocati in modo da connettersi ad alcuni pin predefiniti multifunction che si trovano nel banco 1 e banco 3; quest'ultimo è consigliato per avere una maggiore flessibilità. Prima di scegliere la memoria DDR, si è guardato lo schematico della spartan6 SP605 in quanto essa già possiede una memoria DDR connessa all'FPGA e lo schematico mostra una DDR3 della Microchip con tutte le connessioni necessarie al suo funzio-

namento. Volendo evitare ogni sorta di problemi e possibili errori, si è deciso di optare per quella memoria e replicare lo schematico, prestando attenzione alla connessione con l’FPGA in quanto quella sulla SP605 è una LX45T che non è compatibile in termini di pinout con la LX45. La memoria utilizzata è la MT41J64M16LA-187E della Micron [19], dove 64M16 indica 64Mega parole da 16bit per un totale di 1Gbit mentre LA indica il tipo di package.

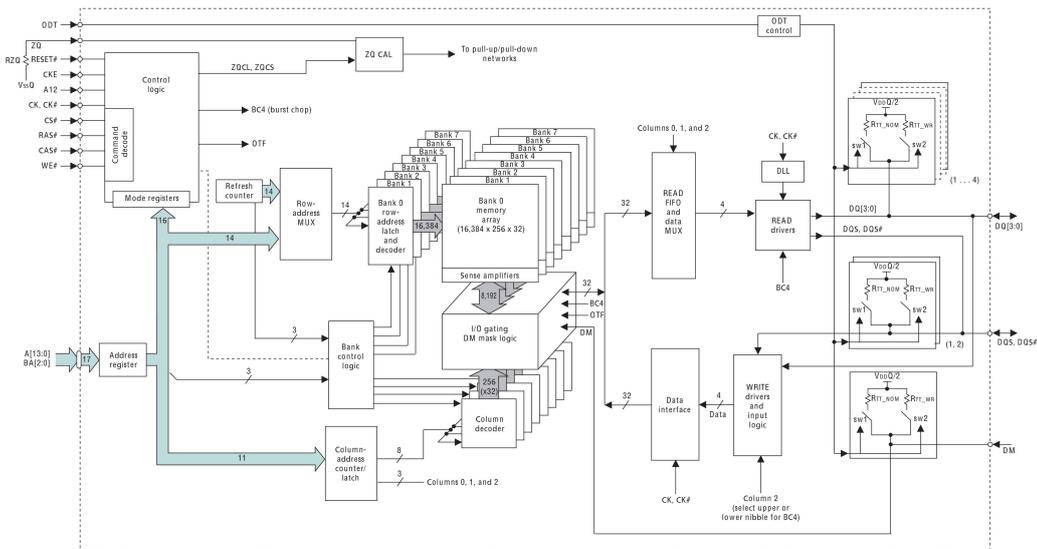


Figura 2.9: Schema a blocchi della memoria DDR

Avendo le DDR3 diversi standard di velocità, le frequenze di funzionamento vengono indicate nell’ultima parte della sigla: 187E indica una DD3-1066. Questa memoria si trova in un package 96-ball BGA e nel caso non fosse disponibile sul mercato allora può essere sostituita con la MT41J64M16JT, che possiede un package leggermente più piccolo, ma con identico footprint per quanto riguarda l’array delle connessioni.

2.3.3 FIFO

Per permettere la comunicazione tra FPGA e Controller USB è necessario interporre una memoria di tipo FIFO tra i due dispositivi. La FIFO utilizzata

è la CYF0018V della Cypress Semiconductor [18] ed il suo studio di questa memoria si è limitato all'analisi dei suoi pin di I/O al fine di interfacciarla con la FPGA.

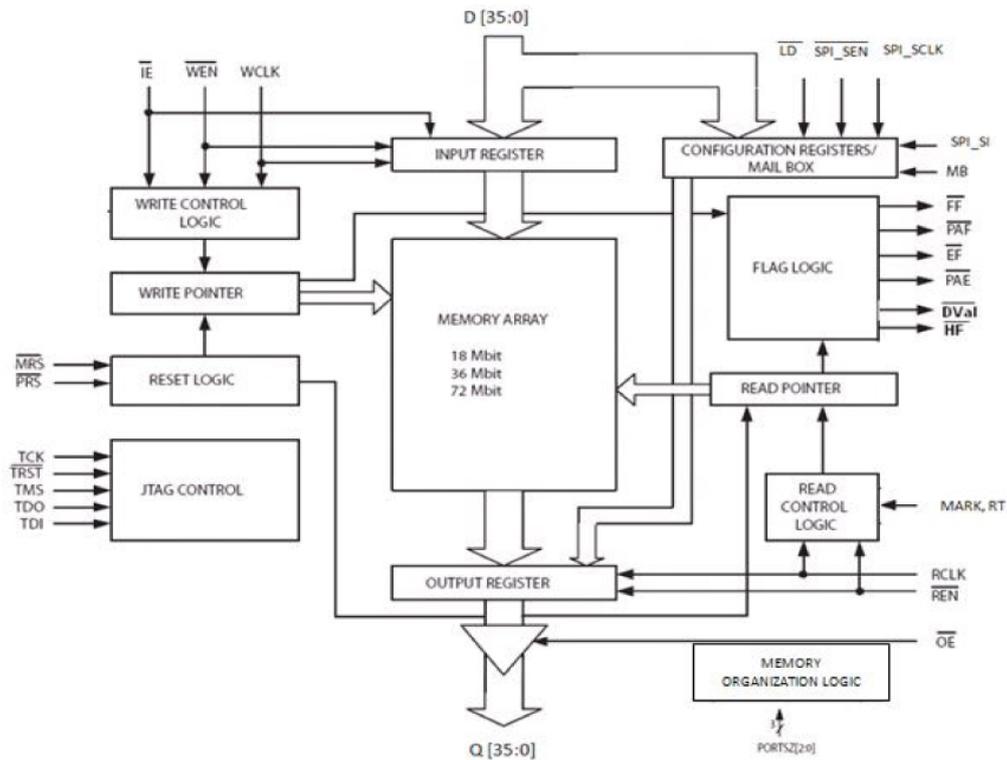


Figura 2.10: Schema a blocchi della memoria FIFO

La famiglia di dispositivi CYF00xxV ha densità di memoria di 18, 36 e 72 Mbit operanti ad una frequenza massima di 133MHz; supporta I/O standard LVC MOS18 ed LVC MOS33 e può essere programmata tramite una connessione seriale I2C o una connessione parallela. Interessante notare che i pin di alimentazione hanno voltaggi differenti ovvero 3.3V, 1.8V, 1.5V e Vref (0.75V); tutte queste alimentazioni dovranno essere tenute presenti durante la progettazione del circuito di alimentazione del sistema. I segnali per la programmazione dovranno essere statici o arrivare dalla FPGA a seconda dell'utilizzo della FIFO, come ad esempio i PORTSZ[2:0] che programmano la FIFO in modo che abbia in ingresso e uscita un certo numero di pin di dato attivi. Il segnale SPISEN

sarà posto a livello alto poiché, essendo attivo basso, non bisogna abilitare la comunicazione I2C, mentre altri pin come IE e OE (input enable e output enable) verranno connessi a massa in quanto attivi bassi. I dati in ingresso, 16 in questo caso, insieme ad altri segnali di controllo, saranno collegati all’FPGA, mentre i 16 in uscita dovranno arrivare ad un connettore per poi passare alla scheda con il controller USB.

2.4 Periferiche

In questa sezione parleremo delle periferiche introdotte nel sistema al fine di svolgere determinati compiti, come per esempio un generatore di clock, un circuito per la gestione del reset o un piccolo integrato per proteggere il software di programmazione.

2.4.1 Clock

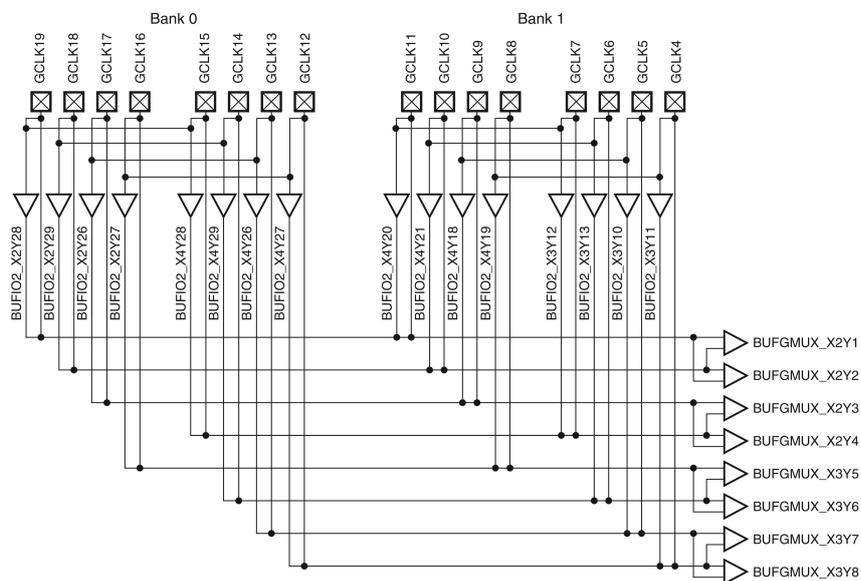


Figura 2.11: Connessione BUFGMUX nel banco 0 e 1

Le FPGA della serie Spartan 6 possiedono al loro interno un sistema di distribuzione del clock sia verso i suoi componenti interni che verso i pin di I/O [9]. Un clock esterno può essere collegato all’FPGA al fine di ottenere poi, tramite circuiti specifici, i clock alla frequenza desiderata.

BUFGMUX Routing Restrictions		Bank 0	Bank 1
BUFGMUX_X2Y1 (I0) BUFGMUX_X2Y2 (I1)	Direct Routing	GCLK_19	GCLK_11
	Indirect BUFIO2	GCLK_19 <BUFIO2_X2Y28>	GCLK_11 <BUFIO2_X4Y20>
	Indirect BUFIO2	GCLK_15 <BUFIO2_X2Y28>	GCLK_7 <BUFIO2_X4Y20>
BUFGMUX_X2Y2 (I0) BUFGMUX_X2Y1 (I1)	Direct Routing	GCLK_18	GCLK_10
	Indirect BUFIO2	GCLK_18 <BUFIO2_X2Y29>	GCLK_10 <BUFIO2_X4Y21>
	Indirect BUFIO2	GCLK_14 <BUFIO2_X2Y29>	GCLK_6 <BUFIO2_X4Y21>
BUFGMUX_X2Y3 (I0) BUFGMUX_X2Y4 (I1)	Direct Routing	GCLK_17	GCLK_9
	Indirect BUFIO2	GCLK_17 <BUFIO2_X2Y26>	GCLK_9 <BUFIO2_X4Y18>
	Indirect BUFIO2	GCLK_13 <BUFIO2_X2Y26>	GCLK_5 <BUFIO2_X4Y18>
BUFGMUX_X2Y4 (I0) BUFGMUX_X2Y3 (I1)	Direct Routing	GCLK_15	GCLK_7
	Indirect BUFIO2	GCLK_15 <BUFIO2_X4Y28>	GCLK_7 <BUFIO2_X3Y12>
	Indirect BUFIO2	GCLK_19 <BUFIO2_X4Y28>	GCLK_11 <BUFIO2_X3Y12>
BUFGMUX_X3Y5 (I0) BUFGMUX_X3Y6 (I1)	Direct Routing	GCLK_16	GCLK_8
	Indirect BUFIO2	GCLK_16 <BUFIO2_X2Y27>	GCLK_8 <BUFIO2_X4Y19>
	Indirect BUFIO2	GCLK_12 <BUFIO2_X2Y27>	GCLK_4 <BUFIO2_X4Y19>
BUFGMUX_X3Y6 (I0) BUFGMUX_X3Y5 (I1)	Direct Routing	GCLK_14	GCLK_6
	Indirect BUFIO2	GCLK_14 <BUFIO2_X4Y29>	GCLK_6 <BUFIO2_X3Y13>
	Indirect BUFIO2	GCLK_18 <BUFIO2_X4Y29>	GCLK_10 <BUFIO2_X3Y13>
BUFGMUX_X3Y7 (I0) BUFGMUX_X3Y8 (I1)	Direct Routing	GCLK_13	GCLK_5
	Indirect BUFIO2	GCLK_13 <BUFIO2_X4Y26>	GCLK_5 <BUFIO2_X3Y10>
	Indirect BUFIO2	GCLK_17 <BUFIO2_X4Y26>	GCLK_9 <BUFIO2_X3Y10>
BUFGMUX_X3Y8 (I0) BUFGMUX_X3Y7 (I1)	Direct Routing	GCLK_12	GCLK_4
	Indirect BUFIO2	GCLK_12 <BUFIO2_X4Y27>	GCLK_4 <BUFIO2_X3Y11>
	Indirect BUFIO2	GCLK_16 <BUFIO2_X4Y27>	GCLK_8 <BUFIO2_X3Y11>

Figura 2.12: Risorse del clock globale condivise tra banchi 0 e 1

Prima di ricercare i componenti adatti alla generazione del clock, si è analizzato il datasheet dell’FPGA inerente alla gestione del Clock: sono presenti numerosi pin di I/O dedicati al Clock, tuttavia non sono indipendenti tra loro

in quanto vi sono dei multiplexer interni collegati a vari pin di clock come mostrato nella figura 2.11. Schema e tabelle simili sono presenti anche per il banco 2 e 3. Ci sono quindi delle restrizioni da rispettare per non avere conflitti sulla distribuzione del clock e nella figura 2.12 vengono mostrate le reti condivise: per esempio se si utilizza il GCLK19 sul banco 0 non sarà possibile utilizzare il GCLK11 sul banco 1 poiché vi sarà un conflitto. Internamente all'FPGA ci sono dei blocchi dedicati chiamati CMT, ovvero Clock Management Tiles, ciascuno dei quali contiene al suo interno due DCM (Digital Clock Managers) e un PLL. Essi sono collegati alla rete di Clock e servono per generare internamente una frequenza voluta a partire da una già esistente. A seconda del dispositivo all'interno ci sono diversi CMT e nel caso della LX45 ce ne sono 4 per un totale di 8 DCM e 4 PLL; dalla LX75 in poi le FPGA possiedono 6 CMT. Queste informazioni saranno utili in fase di design dello schema per non incorrere in conflitti con i clock provenienti dai vari integrati. Per quanto riguarda la scelta della frequenza del clock, si evince dai documenti che la frequenza del clock minima richiesta per il PLL è di 19 MHz; questo sarà il limite minimo di frequenza fornita dal generatore esterno. Quest'ultimo dovrà avere un anche la fondamentale proprietà di essere facilmente reperibile. La prima ricerca è stata fatta analizzando lo schematico della scheda SP605 dove è stato trovato un generatore di clock della Silicon Labs chiamato SI500D [25]. Questo generatore funziona senza l'ausilio di componenti esterni e genera una frequenza fissa variabile a seconda della programmazione effettuata dalla casa costruttrice; in questo caso lavora a 50 MHz. Questo tipo di integrato purtroppo non è facilmente reperibile pertanto è scartato a favore di un dispositivo con quarzo esterno: CDCS502 [17]. L'integrato in questione lavora con una tensione di alimentazione di 3.3V e funziona con cristalli aventi frequenza compresa tra 8 e 32 MHz; un PLL interno può moltiplicare (x4) o meno il segnale di ingresso a seconda dei valori di tensione presenti su alcuni pin di programmazione. Tramite questi ingressi è anche possibile attivare o meno lo Spread Spectrum Clocking che serve a ridurre le emissioni EMI. Per quanto riguarda la scelta del quarzo, ne vengono indicati alcuni tuttavia si è stabilito di utilizzare un dispositivo ibrido; il 357L è perfetto dato che genera una frequenza a 27 MHz ed è facilmente reperibile [15].

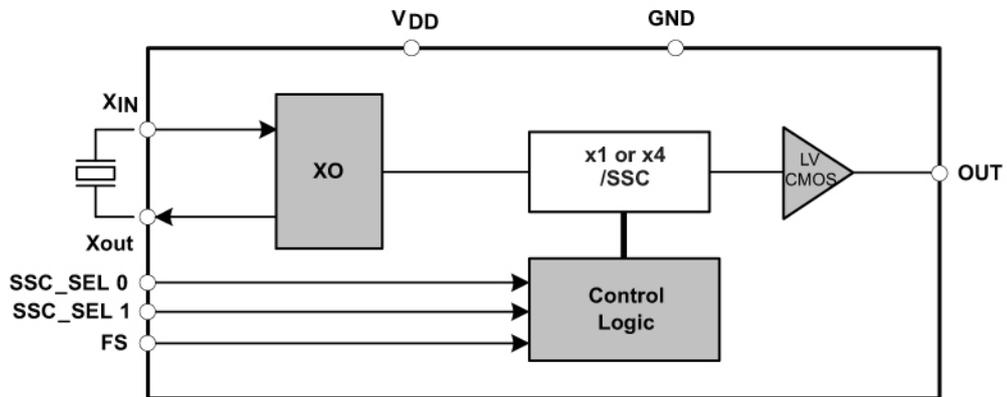


Figura 2.13: Schema a blocchi del circuito di generazione del Clock

2.4.2 Reset

Durante l'accensione il sistema deve essere avviato tramite un segnale di reset generato da un opportuno dispositivo. Questo dovrà avere la caratteristica di poter generare anche l'impulso durante il normale funzionamento al momento dell'attivazione un pulsante; un circuito antirimbalo sarà d'obbligo affinché non vengano generati tantissimi impulsi consecutivi. L'integrato scelto, che dovrà essere collegato alla linea di reset, è il MAX811/812 della Maxim-IC [24] (811 ha reset attivo basso mentre 812 ha reset attivo alto); esso genera il segnale di reset una volta che la tensione di alimentazione raggiunge una data soglia definita dal modello. Questo piccolo dispositivo non richiede di per sé componenti aggiuntivi e ha consumi ridotti tuttavia un pin dedicato, tenuto alla tensione di alimentazione da una resistenza di pull-up interna (20 Kohm) a cui è connesso un circuito antirimbalo, permette di collegare a massa un pulsante per il reset manuale.

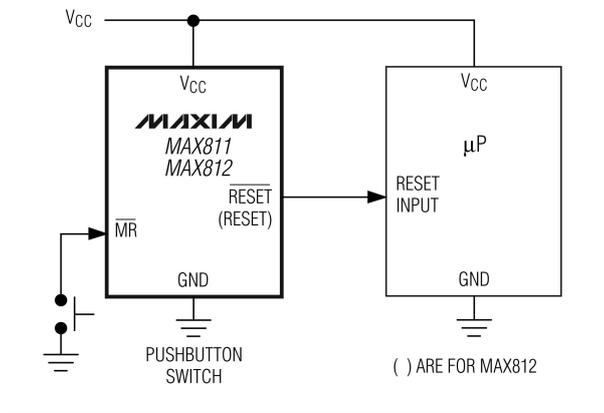


Figura 2.14: Applicazione tipica del Max811/812

Nel nostro caso l'alimentazione del Core della FPGA funziona ad 1.2V, pertanto la soglia di 2.93 è accettabile, ma per incrementare ulteriormente il ritardo, se la tensione di alimentazione sale velocemente, si collegherà in parallelo al pulsante un condensatore (1uF) in modo da fornire il reset per un periodo maggiore durante l'accensione.

2.4.3 Protezione

Per impedire a terze parti di poter leggere i dati di configurazione dell'FPGA occorre trovare un metodo di protezione dei dati contro queste intrusioni. La Xilinx prevede due metodi per proteggere il codice, utilizzando un dispositivo interno all'FPGA o uno esterno. La Spartan6 LX75 e i modelli superiori possiedono al suo interno un sistema di decriptazione (Advanced Encryption Standard) che senza la conoscenza della parola chiave impedisce a chiunque la comprensione del codice, in particolare il sistema AES della spartan 6 consiste in una criptazione del bitstream di configurazione attraverso una parola chiave segreta che avviene a livello software e una decriptaggio del bitstream sul chip sempre grazie alla stessa parola salvata in una memoria dedicata interna all'FPGA. Tale chiave può essere caricata solo attraverso la porta JTAG e non può più essere letta. Vi sono alcuni pin della FPGA dedicati a questo tipo di programmazione, tuttavia la LX45 non supporta questo metodo di protezione: si dovrà dunque prevedere l'utilizzo dell'AES, ma anche un ulteriore sistema di

protezione dati. L'altro metodo necessita di un dispositivo esterno: il principio di funzionamento consiste in un sistema di autenticazione ed è mostrato in figura 2.15 [12]. Sulla FPGA e sul circuito integrato vengono memorizzate due parole chiave identiche (conosciute solo dal costruttore) e l'FPGA invia un codice random al dispositivo esterno. Questo cripta il messaggio con la parola chiave e rispedisce il codice criptato all'FPGA che lo confronterà con lo stesso codice anch'esso criptato internamente. Se i due codici sono uguali, allora l'FPGA verrà abilitato e comincerà a funzionare.

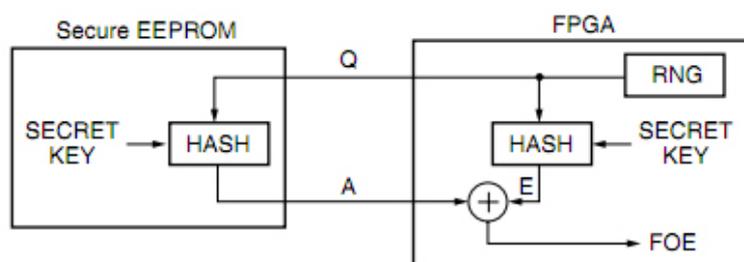


Figura 2.15: Principio di funzionamento del sistema di protezione

Essendo segnalato sui datasheet un componente adatto allo scopo, si è deciso di utilizzarlo onde evitare problemi (DS 2432 della Maxim-IC [21]). Questo piccolo integrato ha la particolarità di avere soltanto due pin: uno connesso a massa e l'altro connesso all'FPGA. Grazie ad una resistenza di pull-up esterna, questa connessione è mantenuta a livello alto e il DS2432 si alimenta proprio da questo pin quando non vi è comunicazione.

2.5 Alimentazione

Le tensioni alimentazione del sistema complessivo devono essere generate a partire da una alimentazione generale di 12V; le tensioni in gioco sono 3.3V, 1.8V, 1.5V, 1.2V e 0.75V. La scelta dei convertitori deve essere fatta in base al consumo di potenza dei dispositivi che dovranno alimentare; il componente che consuma la quantità maggiore di energia è indubbiamente l'FPGA. Purtroppo gli unici dati disponibili riguardano il suo consumo massimo e sono riportati nella figura 2.16.

Power Requirements for Spartan-6 Series							
	Pin Name	Voltage (V)	Imax*		Tolerance	Sequencing Order	Timing Delay
			Device	(mA)			
Core	Vccint	1.21	XC6SLX4 XC6SLX9 XC6SLX16 XC6SLX25 XC6SLX25T XC6SLX45 XC6SLX45T XC6SLX100 XC6SLX100T XC6SLX150 XC6SLX150T	300 500 500 600 600 800 800 1500 1500 3000 3000	±5%		
I/O	Vcco	1.2 / 1.5 / 1.8 / 2.5 / 3.3	X XC6SLX4 XC6SLX9 XC6SLX16 XC6SLX25 XC6SLX25T XC6SLX45 XC6SLX45T XC6SLX100 XC6SLX100T XC6SLX150 XC6SLX150T	1000 1000 1000 1000 1000 1500 1500 2000 2000 3000 3000	±5%	1	Ramp Time (Vccint, Vcco, Vccaux) : 0.2 to 50ms (-1L is 0.2 to 40 ms)
I/O	Vccaux	2.5 / 3.3	All Devices	2000	±5%		

Figura 2.16: Consumi massimi

I valori di corrente richiesta sono esageratamente alti e si riferiscono al caso limite in cui la FPGA lavora al massimo delle sue possibilità; essendo questo dispositivo programmabile può avere dei consumi bassissimi o altissimi a seconda della funzione che svolge, quindi a priori non è facile stimare la potenza richiesta per il suo funzionamento, ma difficilmente si raggiungerà un tale consumo di potenza. La casa costruttrice mette a disposizione un tool per il calcolo della potenza assorbita (XPower Estimator), ma per utilizzarlo occorre avere già il software finale ed in questo momento non è ancora stato ultimato. Si è deciso di istanziare un primo circuito integrato che eroghi 8V dalla alimentazione al fine di poter alimentare anche le altre schede del sistema, successivamente un convertitore che abbassi la tensione a 3.3V da cui generare poi tutte le altre. Al fine di avere la compatibilità anche con dispositivi FPGA dalle prestazioni superiori, si è deciso di sovradimensionare i circuiti di alimentazione, in particolare il primo convertitore sarà switching ed erogherà un massimo di 8A, il secondo sarà un convertitore LDO che potrà fornire un massimo di 7.5A; il convertitore che genera la 1.2V deve alimentare unicamente il Core della FPGA pertanto è stato anch'esso sovradimensionato e dovrà erogare al massimo 3A, mentre gli altri devono quasi tutti generare tensioni di riferimento o poco più quindi saranno meno potenti. Si è innanzitutto guardato alla sezione di alimentazione del sistema già presente in cui era presente un convertitore switching DC-DC dalle caratteristiche decisamente inferiori a quanto richiesto, quindi si sono fatte ricerche su internet fino ad arrivare al

convertitore LTC3608 della Linear Technology [23].

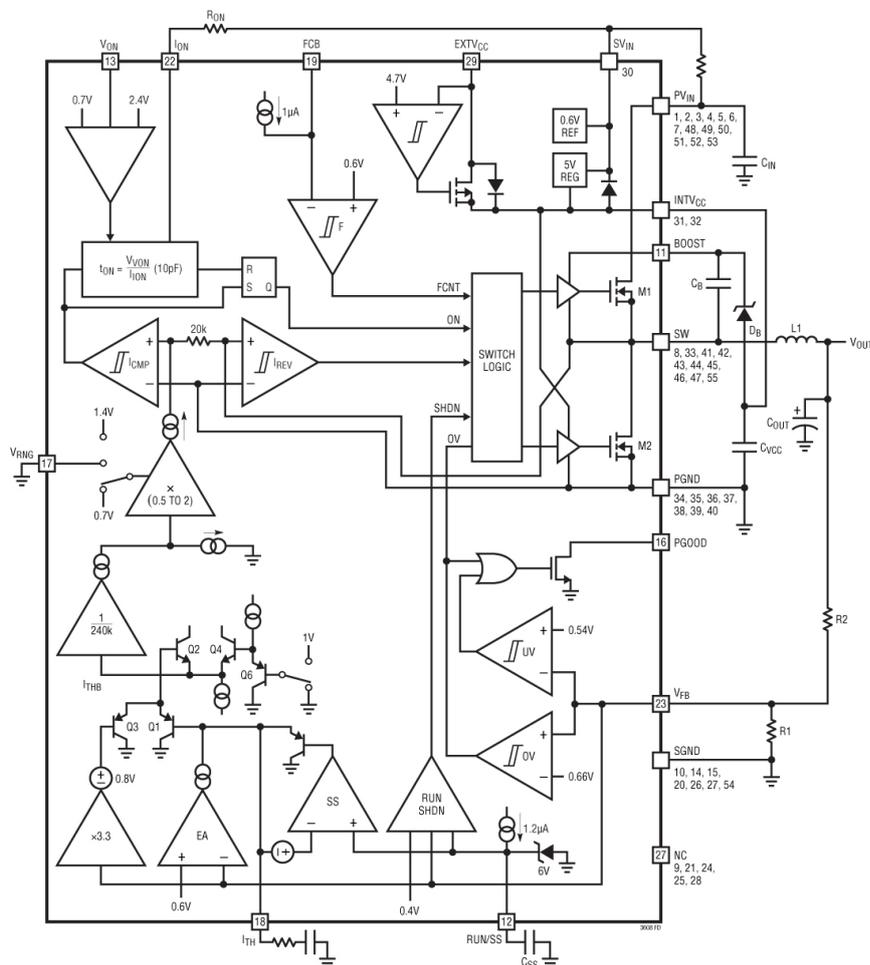


Figura 2.17: Schema interno del convertitore switching

Questo convertitore switching ammette una tensione di ingresso compresa tra 4 e 18V, una corrente di uscita massima di 8A e integra al suo interno i transistori N-channel MOSFET di potenza. Per incrementare la dissipazione di potenza sono presenti, sul lato inferiore del componente, tre larghi pad connessi rispettivamente all'ingresso, uscita e massa. Seguendo le indicazioni sul datasheet, sono state dimensionate le resistenze e l'induttore per la generazione

della tensione voluta in uscita (8V). Le note presenti riguardo ai condensatori sono state annotate al fine di creare un migliore layout. Siccome le potenze prese in considerazione sono le peggiori si è deciso di poter utilizzare diversi induttori al fine di poter far circolare più o meno corrente; avendo questi induttori pinout differenti tra loro, il layout dovrà essere creato in modo da poter ospitare ogni tipologia di induttore indicata. Il convertitore successivo è un LDO per abbassare la tensione a 3V; ancora una volta la scelta si è orientata su un dispositivo della Linear Technology (anche perché sono facilmente reperibili): l'LT1083 [22]. Questo dispositivo ha una tensione di dropout di circa 1V, compatibile con le nostre specifiche di ingresso-uscita, l'uscita è programmabile tramite due resistenze ed è in grado di erogare fino a 7.5A. I package disponibili sono tre, più un quarto ormai obsoleto, ciascuno dei quali possiede prestazioni termiche differenti; il problema infatti è la dissipazione di potenza: questo integrato dovrà dissipare una potenza massima di 37.5W dovuto alla differenza di potenziale tra ingresso e uscita moltiplicato per la corrente massima erogata. Bisogna ancora ricordare che certamente non si raggiungeranno correnti di questa entità, tuttavia la dissipazione di potenza deve esserci e richiede l'utilizzo di un dissipatore.

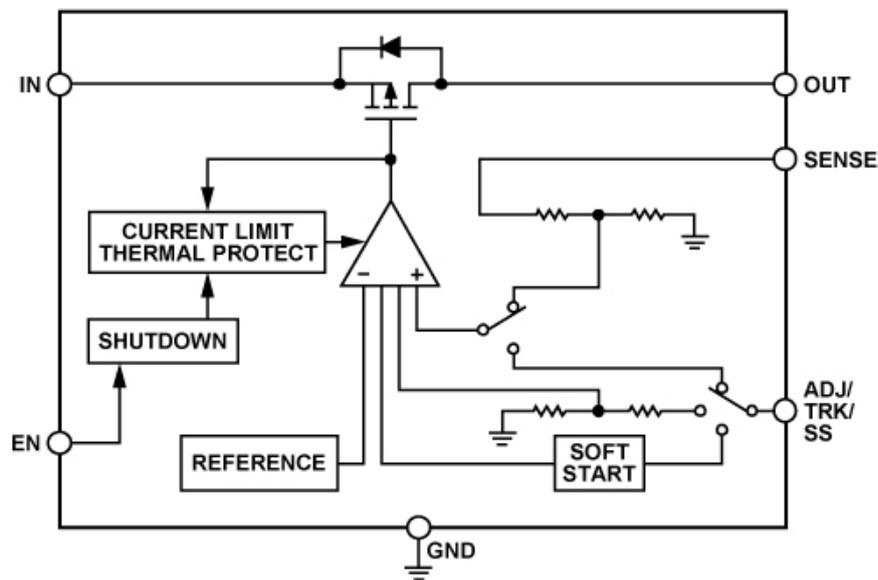


Figura 2.18: Schema interno del convertitore LDO ADP1706

Per non occupare eccessivo spazio il package utilizzato sarà il TO-3P che ha un montaggio verticale; il dissipatore, per garantire un corretto funzionamento con temperatura ambiente a 75 C, dovrà avere una resistenza termica di 0.5C/W quindi di dimensioni decisamente grandi (sempre considerando una corrente di uscita di 7.5A). Per la generazione delle tensioni di 1.8V, 1.5V e 0.75V si sono cercati dispositivi con tensione di uscita fissata a quei valori e la Analog Devices offre una serie di circuiti integrati che svolgono proprio questo lavoro. Gli ADP1706 [16] sono convertitori che operano con tensioni di ingresso comprese tra 2.5V e 5V ed erogano al massimo 1A; sono disponibili in package anche molto piccoli (3mm x 3mm) e sono disponibili 16 versioni che erogano ciascuna un voltaggio prestabilito (16 valori tra 0.75V e 3.3V). Questi piccoli componenti sono dotati anche di un pin per il Soft Start che, durante l'accensione, permette di non avere bruschi cambiamenti di corrente aumentando gradatamente la tensione, ma la cosa più importante è che sono disponibili nelle tensioni desiderate (1.8V, 1.5V e 0.75V). Per alimentare il core della FPGA sono necessari invece 1.2V che saranno generati a partire da 3.3V; essendo la corrente massima assorbita di qualche Amper si utilizzerà anche in questo caso un convertitore lineare con tensione di uscita prefissata a 1.2V e corrente massima di 3A della National Semiconductor: LP38859 [20]. È stato infine inserito un ultimo dispositivo trovato sullo schematico della SP605: il TPS51200 [26] è un regolatore appositamente creato per generare le tensioni per le DDR (Sink/Source DDR Termination Regulator); è dotato di un pin di enable e un pin per il monitoraggio della tensione di uscita. Integra al suo interno il sistema soft-start già visto per gli ADP1706 e possiede un particolare package avente un grande pad al centro per la dissipazione termica. Come per la DDR lo schema verrà copiato al fine di non incorrere in problemi indesiderati.

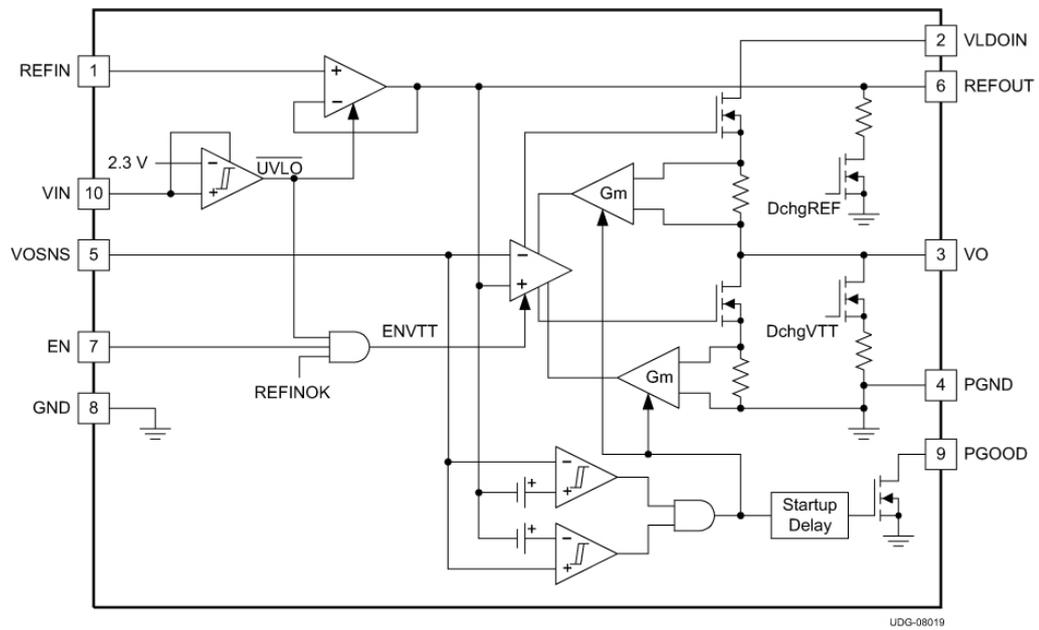


Figura 2.19: Schema interno del DDR SINK/SOURCE

Capitolo 3

Realizzazione degli schematici

Dopo aver studiato come utilizzare i vari dispositivi è iniziata la realizzazione dello schematico utilizzando il programma OrCAD v16.2. Dopo aver creato un nuovo progetto, non resta che disegnare tutti gli schematici desiderati istanziando su di essi i vari componenti; nessuno dei componenti precedentemente analizzati è presente nelle librerie di OrCAD, pertanto dovrà essere fatta una nuova libreria all'interno della quale si potranno creare i vari componenti. Compiere questa azione non è molto difficoltoso: click col tasto destro sulla libreria e selezionare "nuovo componente"; a questo punto si dovrà disegnare un simbolo (un rettangolo) e disporre i pin che saranno caratterizzati dal nome, un numero o posizione, e il suo utilizzo (power, input, output...). La creazione del simbolo dell'FPGA però non è affatto rapida da realizzare, considerando il numero elevato di pin. Inoltre volendo far stare il circuito in fogli A4 così che possano essere stampati o letti, fa sì che non sia possibile creare l'FPGA come unico componente poiché troppo grande per poter stare su un unico foglio (i pin nel simbolo possono essere disposti solo nel contorno del rettangolo). Si è deciso quindi di dividere l'FPGA in 6 componenti diversi che convergeranno in un unico componente quando gli verrà associato il layout: saranno rispettivamente i 4 banchi di I/O in aggiunta a un simbolo sul quale vi saranno le alimentazioni generali e uno su cui si trovano alcuni pin dedicati alla programmazione. Sul datasheet relativo al pinout dell'FPGA sono riportati tutti i pin e la loro posizione per il package FG(G)484 che può contenere al suo interno l'LX45, LX75, LX100 o LX150. In questo caso quindi tutti i pin verranno mappati, ma solo alcuni potranno essere connessi poiché, come precedentemente spiegato, al fine di avere compatibilità, si dovranno utilizzare pin comuni a

tutti i dispositivi. Sarà quindi necessario controllare che l'I/O in questione sia utilizzabile ogni volta che si effettua un collegamento. Il vantaggio è che questo simbolo potrà essere riutilizzato nel caso si debba istanziare una nuova FPGA dalle prestazioni migliori. La DDR è disponibile su vari package, ma quello di interesse è il 96-Ball FBGA in quanto è l'unico ad avere il databus a 16 bit: anche in questo caso si dovranno istanziare tutti i pin con il loro nome e la loro posizione. Essendo molto grande, verranno creati due simboli: uno per i segnali e l'altro per le alimentazioni. La memoria FIFO si presenta in un solo package dalle dimensioni non trascurabili (209-Ball FBGA), quindi non c'è altra scelta che scomporre questo dispositivo in due simboli come la memoria DDR. I dispositivi XCFxxP (Flash PROM) sono disponibili in due tipi di package e questa volta si è scelto il TSOP 48-pin. Tutti gli altri circuiti integrati dovranno possedere il loro simbolo in accordo con il loro pinout. Si passa ora alla fase di design dello schematico creando varie schede in modo da rendere leggibile il circuito. Ogni blocco di I/O della FPGA sarà istanziato in una scheda distinta mente per i blocchi di alimentazione e segnali dedicati se ne userà una soltanto. Un foglio sarà dedicato al blocco segnali della memoria FIFO, uno al blocco segnali della memoria DDR, uno ai blocchi di alimentazione delle due memorie appena citate e uno per le due memorie PROM. La parte di alimentazione sarà distribuita su due schede in quanto è costituita da numerosi componenti ed inoltre sarà necessario creare una scheda dedicata al filtraggio della FPGA. Un foglio verrà dedicato ad alcuni componenti periferici mentre in uno o più fogli saranno istanziati i connettori. Saranno aggiunte infine due schede, una delle quali conterrà alcune note generali e l'altra sarà uno schema a blocchi della scheda finale.

3.1 Schema FPGA

Si comincia quindi collocando il banco 0 della FPGA in un foglio di lavoro; per migliorare le prestazioni del sistema e dell'FPGA il datasheet consiglia di tenere raggruppati i segnali appartenenti ad uno stesso bus, per esempio. Considerando che la DDR verrà connessa al Banco 1 o 3, poiché solo questi dispongono del Memory Controller Block, sul banco 0 verrà collegata la memoria FIFO; si scelgono quindi pin tali per cui non ci siano incompatibilità tra i vari dispositivi Spartan 6 e si iniziano a creare i collegamenti tra pin e un simbolo di OFF-PAGE. A questo simbolo verrà dato un nome, che

sarà indicativo anche della net, e sarà presente anche su almeno un altro foglio di lavoro; la net quindi sarà presente su più schede e le net con lo stesso nome saranno connesse tra loro (ovviamente quelle che hanno il riferimento OFF-PAGE). Essendo disponibili solo simboli OFF-PAGE, uno con la freccia a destra e l'altro con la freccia a sinistra, può essere utile creare un nuovo simbolo al fine di indicare un segnale di tipo bidirezionale. Il metodo di creazione è identico a quello per creare un nuovo componente con la differenza che, anziché selezionare "nuovo componente", si dovrà selezionare "nuovo simbolo" quindi scegliere riferimento OFF-PAGE. Per quanto riguarda la memoria FIFO si dovranno scegliere 16 pin di dati e altri 10 pin per la programmazione della memoria e saranno connessi ai loro riferimenti con un nome indicativo del tipo di segnale. Siccome durante il testing del sistema già esistente si sono verificati problemi sulle linee dei segnali, è opportuno inserire delle resistenze di pull-up su tutti questi segnali indicando con una nota che sono resistenze Single In-line; queste potranno essere poi montate o meno. Da notare che è presente anche un pin particolare, HSWAPEN: questo segnale impone delle condizioni di tri-state, pull-up o pull-down su alcuni pin specifici durante la configurazione iniziale dell'FPGA a seconda del suo valore di tensione. Per effettuare delle prove si è scelto di mettere una strip di 3 jumper in modo da poter collegare l'HSWAPEN, che si trova collegato al pin centrale, a massa o a 3.3V attraverso una resistenza. Su questo banco l'alimentazione sarà a 3.3V. Volendo seguire i consigli trovati sul datasheet, la memoria DDR sarà connessa al banco 3 pertanto il banco 1 risulta libero; su questa parte di FPGA faremo convergere numerosi segnali in quanto sappiamo già che il banco 2 avrà pochi I/O disponibili a causa della compatibilità con la LX75. Collegheremo tutti i segnali che arrivano dalla scheda contenente il deserializzatore, che sono i dati provenienti dalle due telecamere oltre ad alcuni segnali di controllo come reset, ora comandato da un microcontroller, e l'I2C. Oltre a questi segnali dobbiamo collegare anche i controlli della schedina con il controller USB; i dati che invece verranno inviati al controller sono le uscite della memoria FIFO. Sempre per i problemi riscontrati con la memoria FIFO sulle linee di interconnessione, vanno posizionate altre resistenze di pull-up sulle linee che collegano la scheda usb e la FPGA indicando che sono resistenze Single In-line. Due segnali provenienti dalla porta di comunicazione sono degli I2C, come anche due segnali provenienti dal deserializzatore: si è deciso di connetterli assieme interponendo però due jumper per poter isolare le due linee incrementando la flessibilità del sistema. È opportuno ricordare che lo standard I2C richiede

Configuration Mode	M[1:0]	Bus Width	CCLK Direction
Master Serial/SPI	01	1, 2, 4 ⁽¹⁾	Output
Master SelectMAP/BPI ⁽²⁾	00	8, 16	Output
JTAG ⁽³⁾	xx	1	Input (TCK)
Slave SelectMAP ⁽²⁾	10	8, 16	Input
Slave Serial ⁽⁴⁾	11	1	Input

Figura 3.1: Modalità di configurazione delle Spartan 6

resistenze di pull-up che sui segnali provenienti dal deserializzatore non sono state applicate; essendo lo schema già molto pieno queste resistenze verranno inserite nello schema dei connettori. Un'altra considerazione riguarda i clock, poiché come analizzato precedentemente, bisogna collegarli su pin che non generino conflitti. Sul banco 0 sono stati collegati due clock al pin GCLK12 e GCLK13. Guardando la figura del precedente capitolo si può facilmente osservare che utilizzando sul banco 1 i pin GCLK8, GCLK9, GCLK10 e GCLK11 si evitano conflitti. I pin di clock provenienti da deserializzatore, che sono 2, e i 2 restanti che arrivano dalla porta USB saranno connessi a queste reti. Onde evitare problemi dovuti ai clock in altre fasi della sperimentazione, si è deciso di connettere USB_CLKOUT e inHIROSE_RCLK_1 anche a due normali pin di I/O. Per terminare questo foglio di lavoro si sono inseriti 9 HEADER (strip di 9 jumper) che fungono da test point; avendo deciso cosa connettere siamo in grado di determinare la tensione di alimentazione degli I/O del banco 1 cioè 3.3V, tensione di alimentazione sia dell'USB che del deserializzatore. Il banco 2 è caratterizzato dalle connessioni con le Flash PROM in quanto possiede i pin dedicati a tali collegamenti. Innanzitutto si sono creati i riferimenti OFF-PAGE per i segnali che termineranno nelle PROM: può essere utile includere nello schema dei test point per questi segnali in modo da monitorare i segnali durante la configurazione iniziale per capire se avviene correttamente. Vi sono anche altri due pin che servono alle modalità di configurazione e in figura A.17 sono mostrati i vari livelli di tensione da applicare per la programmazione.

Utilizzando la modalità Master Serial M1=0 e M0=1 e quindi si devono effettuare le rispettive connessioni sullo schema. Riguardo alla configurazione, nel capitolo precedente si era parlato di poter modificare i contenuti delle memorie PROM e riprogrammare tramite un microcontroller mappato sulla

FPGA che prendeva il controllo dei segnali JTAG. In questo banco 4 pin saranno dedicati ai 4 controlli della porta JTAG oltre ad un quinto pin opzionale per mettere in alta impedenza questi quattro segnali (tramite la programmazione dell'FPGA). Per una maggiore flessibilità si richiedeva di inserire una uscita LVDS; si sono quindi utilizzati quattro pin a due a due differenziali: clock e dati. Tra i due pin differenziali si è aggiunta anche una resistenza eventualmente applicabile in quanto internamente vi è già una resistenza da 100 ohm. Questi quattro segnali dovranno arrivare ad un connettore che sarà posizionato in una scheda connettori. Su questo banco, che sarà alimentato a 3.3V, saranno connessi anche i 3 led e i 3 ingressi per il testing (LED1,2,3 e PROVA 1,2,3); questi componenti però verranno posizionati in un'altra scheda. Sul banco 3 sarà connessa soltanto la memoria DDR dal momento che richiede come tensione di alimentazione 1.5V. Le connessioni sono esattamente quelle riportate nello schematico della SP605 con la differenza che i pin ai quali vanno collegati i segnali provenienti dalla memoria sono in posizioni differenti rispetto a quanto riportato dallo schematico (SP605 monta una LX45T). Il foglio di lavoro successivo è molto semplice: sono istanziati il banco di alimentazione della FPGA e il banco relativo a segnali dedicati; il primo avrà pin che vanno connessi alle rispettive alimentazioni mentre nel secondo i pin della porta JTAG dovranno uscire della scheda per arrivare nella scheda della PROM mentre occorre fare attenzione per gli altri pin. Il segnale SUSPEND in questo progetto non interessa quindi viene connesso a massa, il VFS va connesso a VCCAUX, VBATT e RFUSE servono per la programmazione del AES che sul dispositivo LX45 non è presente. Tuttavia per avere compatibilità si connette la VBATT a VCCAUX tramite un jumper in modo da poterla scollegare per effettuare dei test, così come la RFUSE viene collegata a massa tramite un jumper e una resistenza dal valore particolare indicato sul datasheet. Un'ultima scheda riguarda unicamente il filtraggio delle varie alimentazioni dell'FPGA; i valori e il numero di condensatori sono stati estrapolati dallo schematico della SP605.

3.2 Schema DDR

Il foglio successivo sarà quello della memoria DDR; anche in questo caso si dovrà avere solo l'accortezza di effettuare le connessioni come mostrato nello schematico della SP605 [13]. Per facilitare la realizzazione del layout è stata aggiunta una nota di riferimento alla pagina del datasheet della memoria DDR

contenente alcuni consigli. In schemi di questa complessità con segnali ad alte frequenze conviene seguire le linee guida proposte dal costruttore durante la realizzazione del layout. E' stata aggiunta anche una nota per le resistenze che dovranno essere Single In-line.

3.3 Schema FIFO

La scheda della memoria FIFO sarà provvista del componente avente i segnali e dai riferimenti OFF-PAGE. I segnali che arrivano alla memoria in ingresso e i segnali di controllo di uscita sono tutti connessi all'FPGA sul banco 0 a parte i segnali relativi alla JTAG. I riferimenti dovranno avere lo stesso nome di quelli presenti sul foglio di lavoro dell'FPGA, i dati in uscita invece dovranno terminare sulla scheda USB quindi usciranno da questo foglio di lavoro per rientrare nella scheda connettori. Per avere una maggiore flessibilità la memoria FIFO deve poter essere bypassata quindi tutti i 16 segnali di dato saranno connessi ai 16 pin di dati in uscita tramite una strip di jumper. La programmazione tramite JTAG come anticipato potrà avvenire in modo indipendente rispetto alla programmazione della FPGA quindi verrà utilizzato un connettore JTAG dedicato (i segnali JTAG possiedono una resistenza di pull-up).

3.4 Schema FIFO e DDR: alimentazione

Siccome la memoria FIFO e la memoria DDR sono state suddivise ciascuna in due simboli, si sono istanziati in un'unico foglio di lavoro le parti relative all'alimentazione. Oltre a ciò, sono state posizionate anche i condensatori di filtraggio delle alimentazioni per i rispettivi componenti.

3.5 Schema PROM

Anche questo schema è ricavato dalle connessioni mostrate nel datasheet a proposito della configurazione Master serial daisy chain mostrata nel precedente capitolo. In questo foglio di lavoro sono stati però inseriti anche i condensatori necessari al filtraggio delle alimentazioni di questi dispositivi e i segnali JTAG sono stati connessi ai riferimenti OFF-PAGE relativi alla possibile program-

mazione tramite Microcontroller. Sono stati quindi interposti dei jumper tra i pin di I/O della FPGA e i segnali JTAG.

3.6 Schema periferiche

Qui vengono posizionati i 3 led connessi a massa tramite una resistenza, i 3 pin per il testing composti ciascuno da una strip di tre jumper, il circuito integrato per la protezione del software, provvisto di resistenza di pull-up sul pin di segnale/alimentazione, il controllo per poter mettere in condizioni di alta impedenza i pin di programmazione ausiliari tramite JTAG posti nel banco 2, e il Clock Generator. Quest'ultimo in particolare è stato dotato di una strip di 3 jumper su un pin di programmazione in modo da poter effettuare delle prove in caso di problemi sulla generazione del clock. Nello schema il quarzo è stato rimpiazzato dal circuito oscillatore ibrido 357L.

3.7 Connettori

I connettori da utilizzare dovranno essere complementari a quelli presenti sulla scheda USB e sulla scheda deserializzatore. Quest'ultima comunica tramite una porta chiamata HIROSE FX2 100-pin che dovrà essere replicata su una scheda assieme ad un connettore 40-pin presente sul circuito del controller USB. Sul connettore più grande escono tutti i dati provenienti dalle telecamere e c'è anche una coppia di segnali I2C a cui verranno collegate due resistenze di pull-up (quando si parlava del banco 1 si era detto che sarebbero state messe nelle schede dei connettori). Sempre per evitare problemi che hanno avuto luogo durante i primi testing con la evaluation board, sono state aggiunte, in serie ai dati delle telecamere ed ai segnali di controllo del deserializzatore, delle resistenze Single In-line. Come posto dalle specifiche si era richiesto di valutare la possibilità di bypassare il deserializzatore e avendo trovato la possibilità di effettuare questa operazione si posizioneranno due connettori relativi alle due telecamere per la comunicazione con la FPGA. I segnali che viaggiano sulla I2C tra deserializzatore e l'unità di elaborazione non sono altro che i dati provenienti dai due sensori, dovranno quindi essere collegati assieme. Il circuito di reset, che verrà posizionato a seguire, dovrà resettare la FPGA; per rendere più flessibile il sistema si interporrà un jumper sulla linea di reset per poter isolare il reset generato dal circuito integrato con quello proveniente dal dese-

rializzatore. Su questa scheda di lavoro trova posto anche un connettore per l'uscita LVDS. Essendo una uscita in corrente, una buona soluzione è utilizzare un cavo SATA: in particolare, è stato posizionato un connettore SATA a 7 pin connesso ai segnali LVDS. In un secondo foglio relativo ai connettori è collocato il connettore a 40-pin i cui riferimenti OFF-PAGE sono presenti nella scheda di lavoro del banco 1 e della FIFO. In questo stesso foglio dovrà essere posizionato anche un connettore di uscita al quale poter collegare in futuro una scheda contenente o meno memoria FIFO e USB 3.0 o Ethernet 10/100/1000 in modo da poter avere una larghezza di banda maggiore. Saranno portati sul connettore tutti i segnali di dato e di controllo della memoria FIFO e tutti i segnali per controllare la porta USB (serviranno per poter programmare l'eventuale nuovo controller di comunicazione). Avendo a disposizione molti pin, collegati a massa, si è deciso per alcuni di essi di interporre tra massa e pin un jumper in modo da poterlo collegare o utilizzarlo come test poi togliendo il jumper.

3.8 Alimentazioni generali

Su un primo foglio di lavoro verranno istanziati il convertitore switching e il regolatore dedicato alle DDR. Il convertitore switching è stato connesso secondo le note applicative presenti sul datasheet; questo tipo di convertitori necessita di particolari attenzioni durante la creazione del layout quindi è stata aggiunta una nota informativa a riguardo. Sulla sua uscita è stato posto un jumper per poter scollegare questo tipo di alimentazione e collegarvi una seconda, mentre il segnale +8V è stato portato su un connettore per alimentare altre schede. Su un secondo foglio di lavoro relativo alle alimentazioni sono stati istanziati tutti i dispositivi lineari per la generazione di tutte le altre tensioni di alimentazione. Alcune note informative sono state aggiunte affinché nel layout i componenti che dissipano maggior quantità di potenza vengano posizionati sul bordo della scheda finale in modo da potervi attaccare un dissipatore mentre altre servono per migliorare la comprensione dello schema suggerendo le schede connesse ad una determinata alimentazione. Su questo foglio si trova anche, oltre ai collegamenti tra l'alimentazione generale 3.3V e quelle dedicate ai vari componenti, il piccolo circuito di reset.

3.9 Note e disposizione

Gli ultimi due fogli di lavoro riguardano note informative e la disposizione dei componenti su scheda. Le note servono soprattutto a chi dovrà eseguire il layout poiché forniscono indicazioni sui valori e tolleranze dei componenti distribuiti nel sistema. Le annotazioni devono aiutare la comprensione di tutto lo schematico pertanto dovranno riguardare la maggior parte di componenti: i condensatori per esempio sono per lo più ceramici con dielettrico X5R, quindi non sarà indicato su ogni singolo condensatore, ma sarà aggiunta una nota in cui si avverte che tutti i condensatori sono ceramici X5R dove non specificato altrimenti. La disposizione dei circuiti integrati invece è un semplice disegno che dovrebbe fornire indicazioni generali su come dovrebbe essere la scheda una volta terminato il layout. Tuttavia la disposizione dei componenti esterni all'FPGA è stata arbitraria anche se rispetta più o meno l'orientamento dei banchi e segue il percorso logico dei segnali.

3.10 Schematico

Il risultato del lavoro di progettazione hardware è uno schematico composto da 17 fogli su cui si sviluppa l'intero circuito elettronico. Nell'Appendice A sono presenti gli schematici finali. La scheda ottenuta avrà le seguenti caratteristiche:

- FPGA serie Spartan-6 LX45 (316 I/O pins) / LX75 (280 I/O pins) / LX100 (326 I/O pins) / LX150 (338 I/O pins), package FG(G)484
- 2x Memoria Flash PROM XCF32P da 4MByte ciascuna
- Memoria DDR3, 128MByte, 1066MHz
- Memoria FIFO, 18Mbit, 300MHz
- Circuito di autenticazione tramite integrato DS2432
- Circuito di clock a 27MHz
- Circuito di Power-on reset con manual reset
- Connettore Hirose 100 pin per deserializzatore

- Connettore 40 pin per scheda USB 2.0
- Connettore Hirose 100 pin per scheda USB 3.0 o Ethernet 10/100/1000
- 2xConnettori 10 pin flat per ingressi videocamere
- Convertitore Switching: 12V - 8V
- Convertitori lineari LDO: 8V - 3.3V, 3.3V - 1.8/1.5/1.2/0.75
- Regolatore dedicato per le terminazioni della memoria DDR3.
- Led, Jumper e test point

Al fine di poter programmare l'FPGA necessario associare ad ogni pin, il nome della rete a cui connesso; si è quindi creato un file dall'estensione UCF e dalla struttura prestabilita che esegue questa associazione tra rete e pin: il file si trova nell'Appendice B e prende spunto dall' originale della SP605.

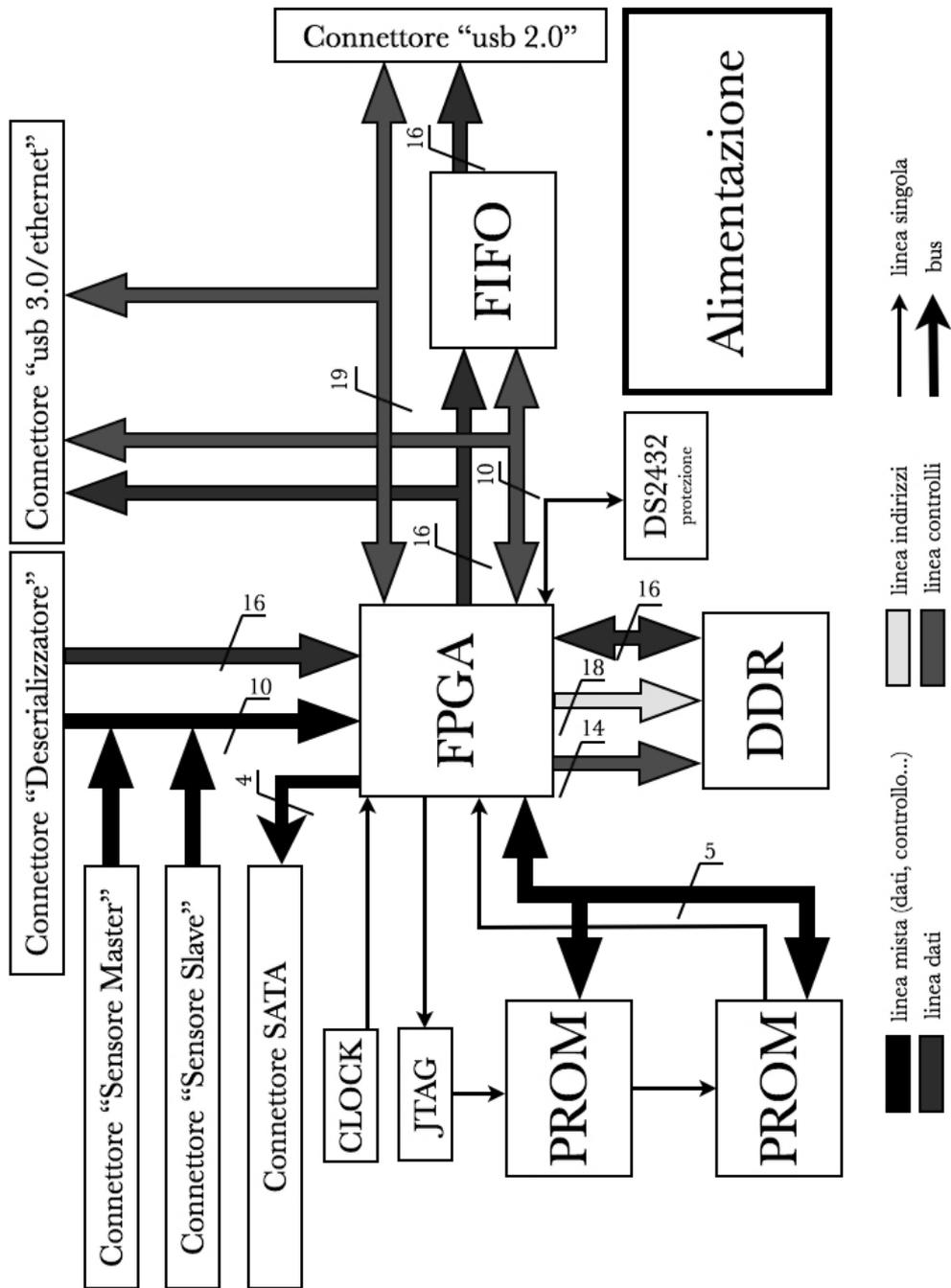


Figura 3.2: Schema a blocchi della scheda da realizzare

Capitolo 4

Conclusioni

4.1 Risultati

Lo studio dei vari componenti ho permesso la realizzazione degli schematici OrCAD presenti nell'appendice; le specifiche della scheda sono state soddisfatte: la scheda che ne risulterà avrà a bordo una FPGA della Xilinx LX45, il cui footprint è compatibile con dispositivi più potenti, integra due memorie PROM per la programmazione, una memoria DDR3 utilizzabile dagli algoritmi che verranno in futuro mappati sulla FPGA, una memoria FIFO, il connettore JTAG per la programmazione, un HIROSE 100 pin per collegare la scheda deserializzatore, un HIROSE 100 pin per collegare un'altra eventuale scheda di interfacciamento, due connettori flat per ricevere i dati delle telecamere direttamente dalle due schede sensori, un connettore 40 pin su cui verrà collegata l'interfaccia USB2.0, un circuito di clock a 27 MHz, un circuito di power-on reset, test point e led di prova, memoria eprom di protezione e circuiteria dedicata all'alimentazione. Dai file OrCAD è stato realizzato il layout della scheda che presenta 8 layer di metalizzazione mostrati nell'appendice C. Alcuni ritardi, dovuti ad errori sullo schema e ai tempi necessari per la sua realizzazione, hanno dilatato i tempi di questa ultima fase che non ha permesso di ottenere la scheda finale in tempi utili per essere collaudata.

4.2 Sviluppi futuri

Innanzitutto sarà necessario verificare che i circuiti di alimentazione funzionino correttamente osservando le diverse tensioni di alimentazione. Successivamente si dovrà concentrare l'attenzione sulle PROM: è molto importante riuscire a caricare sulle memorie i file di configurazione della FPGA in modo che all'accensione quest'ultima venga programmata dalle PROM. Per far ciò si dovrà creare un file VHDL che consenta di verificare il buon fine della programmazione come per esempio l'accensione di alcuni led a seconda degli ingressi nei test point. Verificato che la configurazione avvenga anche senza computer dovranno essere effettuati test sulle varie funzionalità della FPGA come circuiti di clock e contatori; successivamente si passerà a verificare l'interfaccia con la scheda USB 2.0, il circuito di protezione e la memoria DDR3. Una volta stabilita la comunicazione con un Host potranno essere mappati sulla FPGA gli algoritmi necessari all'elaborazione delle immagini video. Si potrà poi provare ad eliminare la scheda deserializzatore implementandolo all'interno della FPGA ed infine trovare un' alternativa al circuito di comunicazione USB 2.0 come ad esempio una scheda con controller USB 3.0 la quale dovrà avere anche una sua memoria FIFO. Una volta terminati tutti i test saranno chiare le scelte da fare per eliminare le problematiche che vi saranno e per arrivare al circuito definitivo eliminando tutti i componenti superflui: l'interfaccia con la scheda deserializzatore potrebbe essere eliminata così come quella per la USB 2.0. La FIFO a questo punto verrà implementata sulla scheda USB 3.0 e quindi potrà essere eliminata. Test point, jumper e resistenze superflue verranno eliminate ed una volta conosciuta la mappatura interna della FPGA sarà possibile effettuare una stima significativa della potenza dissipata dalla FPGA e quindi eventualmente alleggerire la circuiteria di alimentazione. Terminata questa minimalizzazione del sistema si potranno anche rivedere i connettori al fine di migliorare il montaggio rendendolo verticale (una scheda sopra l'altra), diversamente da come avviene ora in cui i connettori HIROSE rendono necessario un montaggio orizzontale.

Appendice A

Schematici

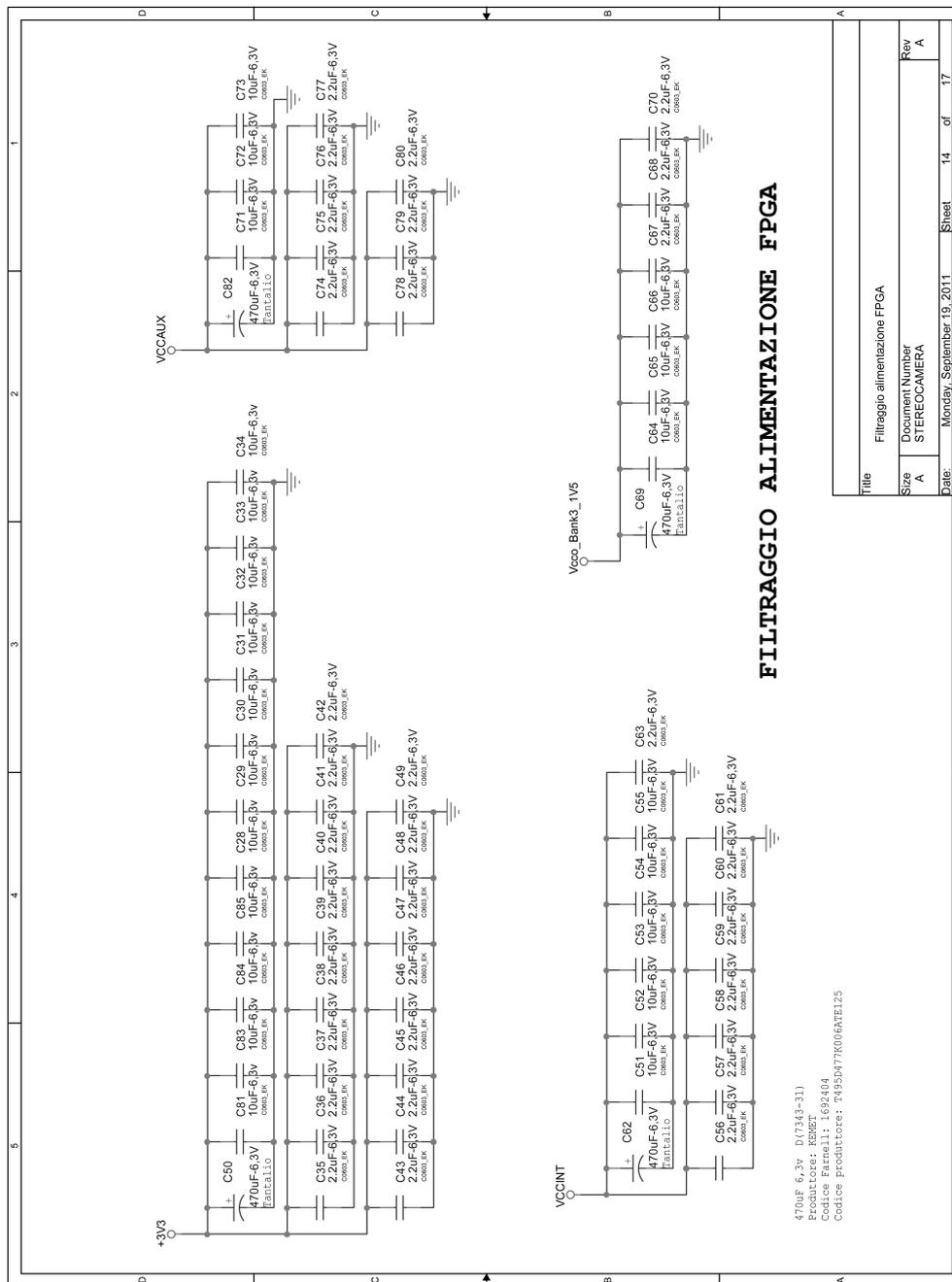


Figura A.3: Filtro di alimentazione FPGA

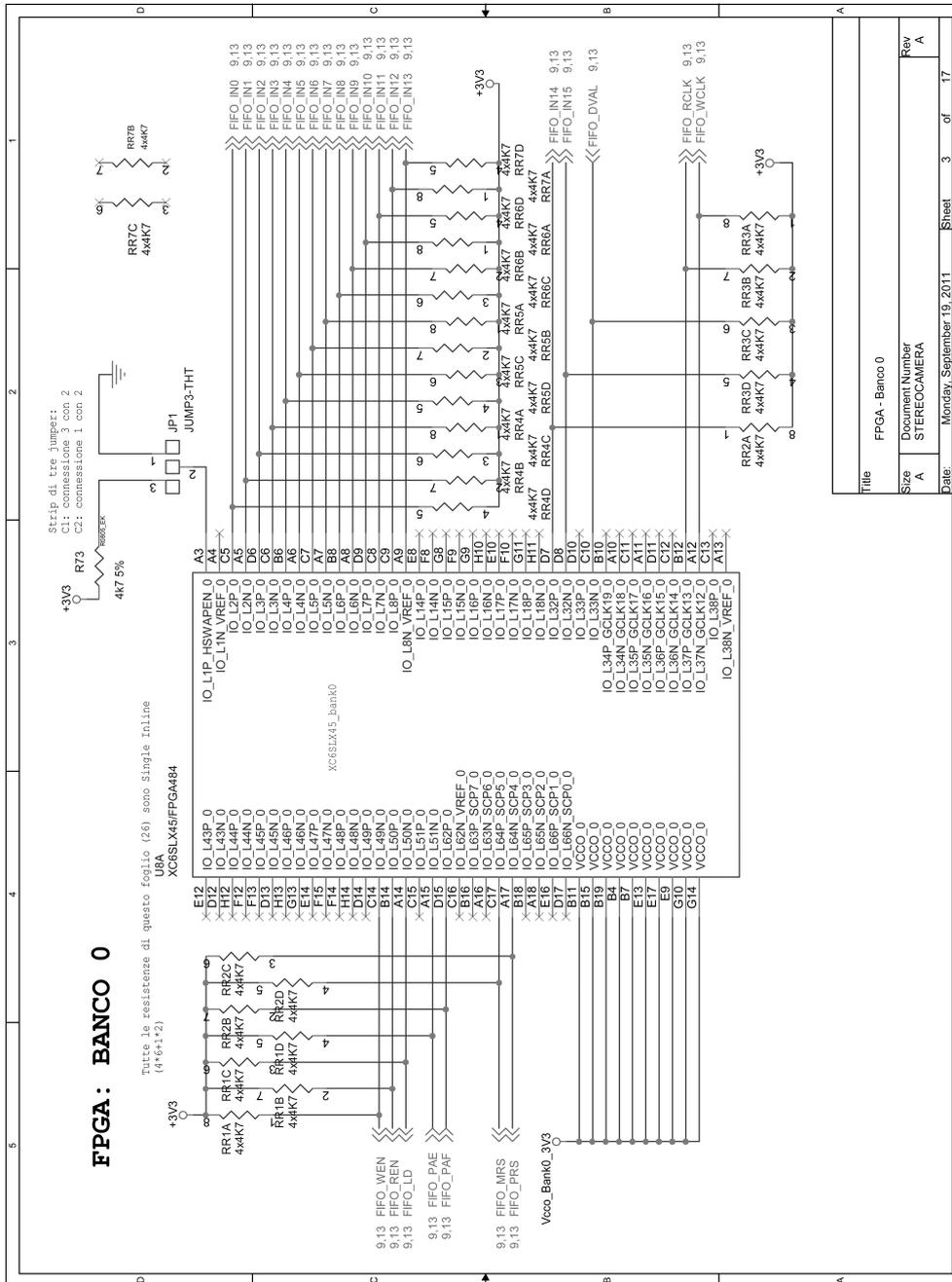


Figura A.4: FPGA: banco 0

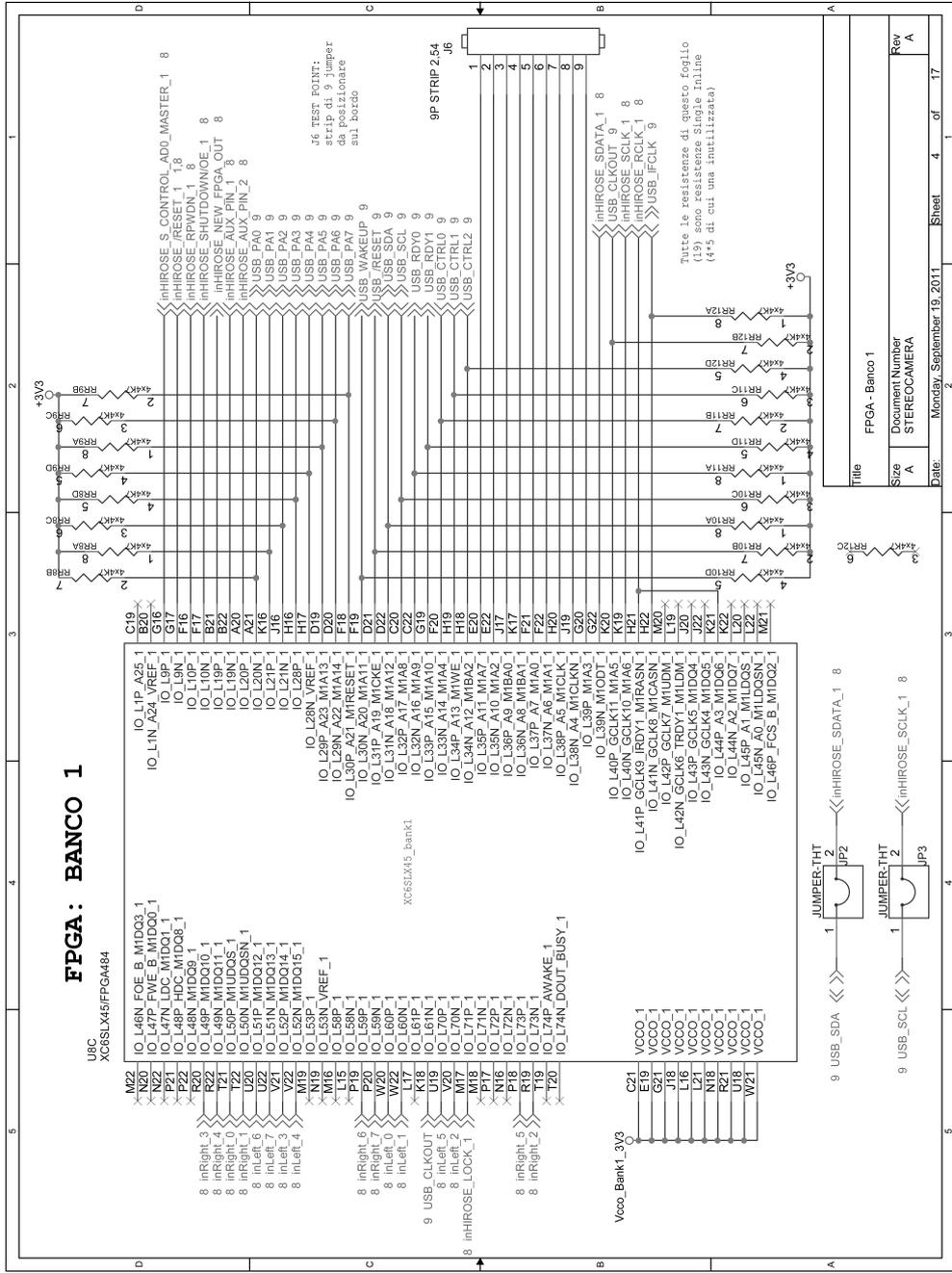


Figura A.5: FPGA: banco 1

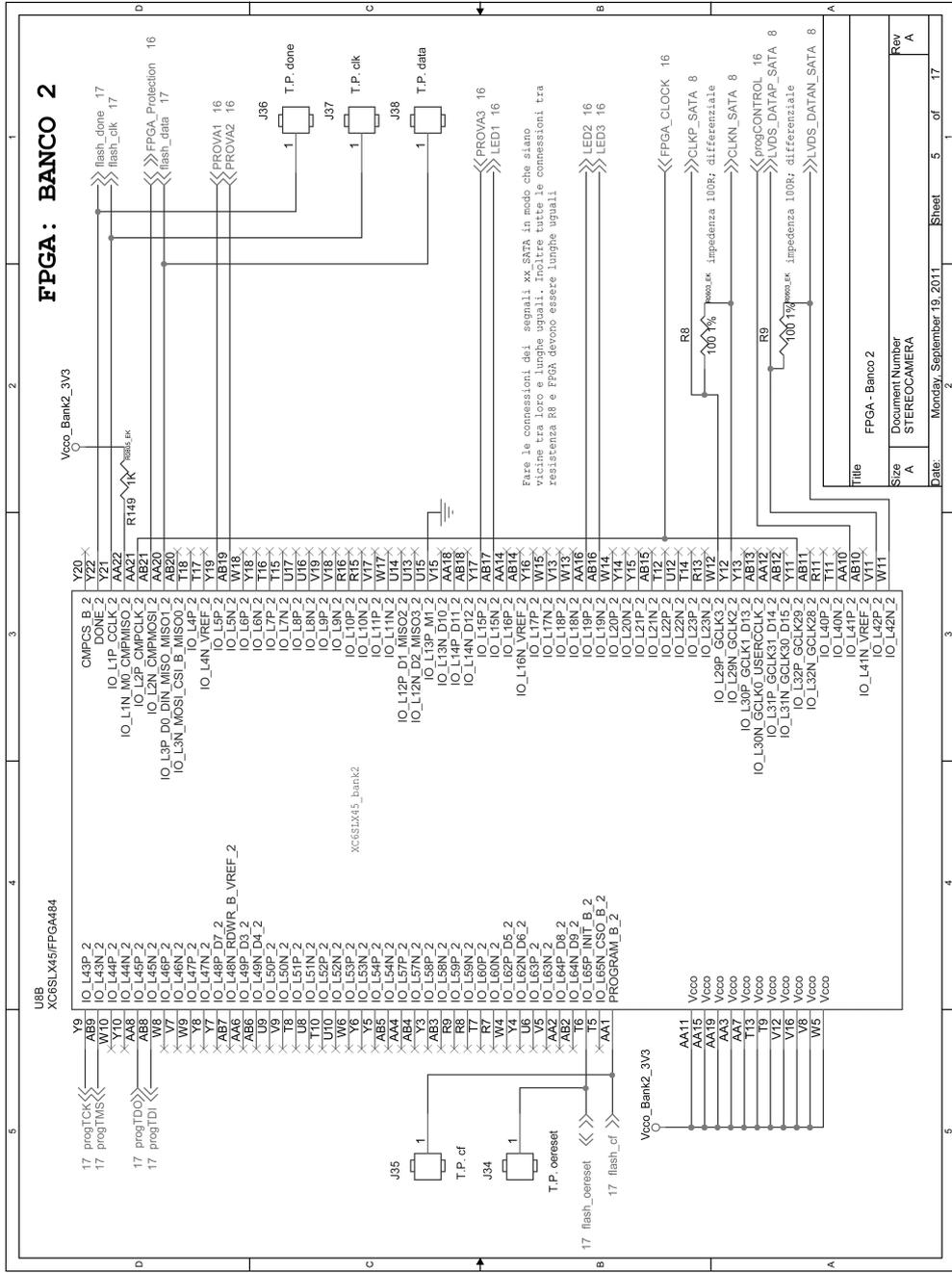


Figura A.6: FPGA: banco 2

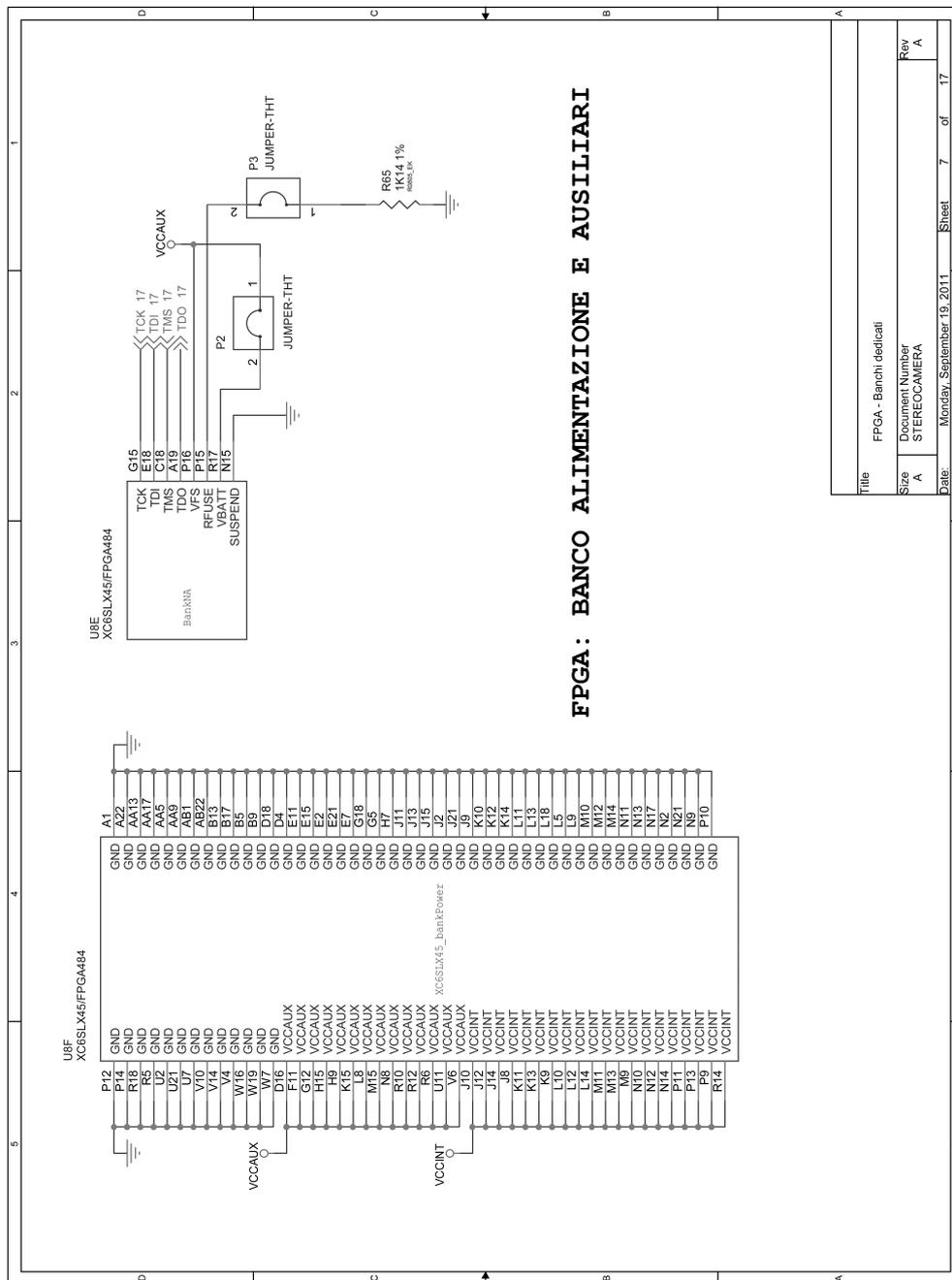
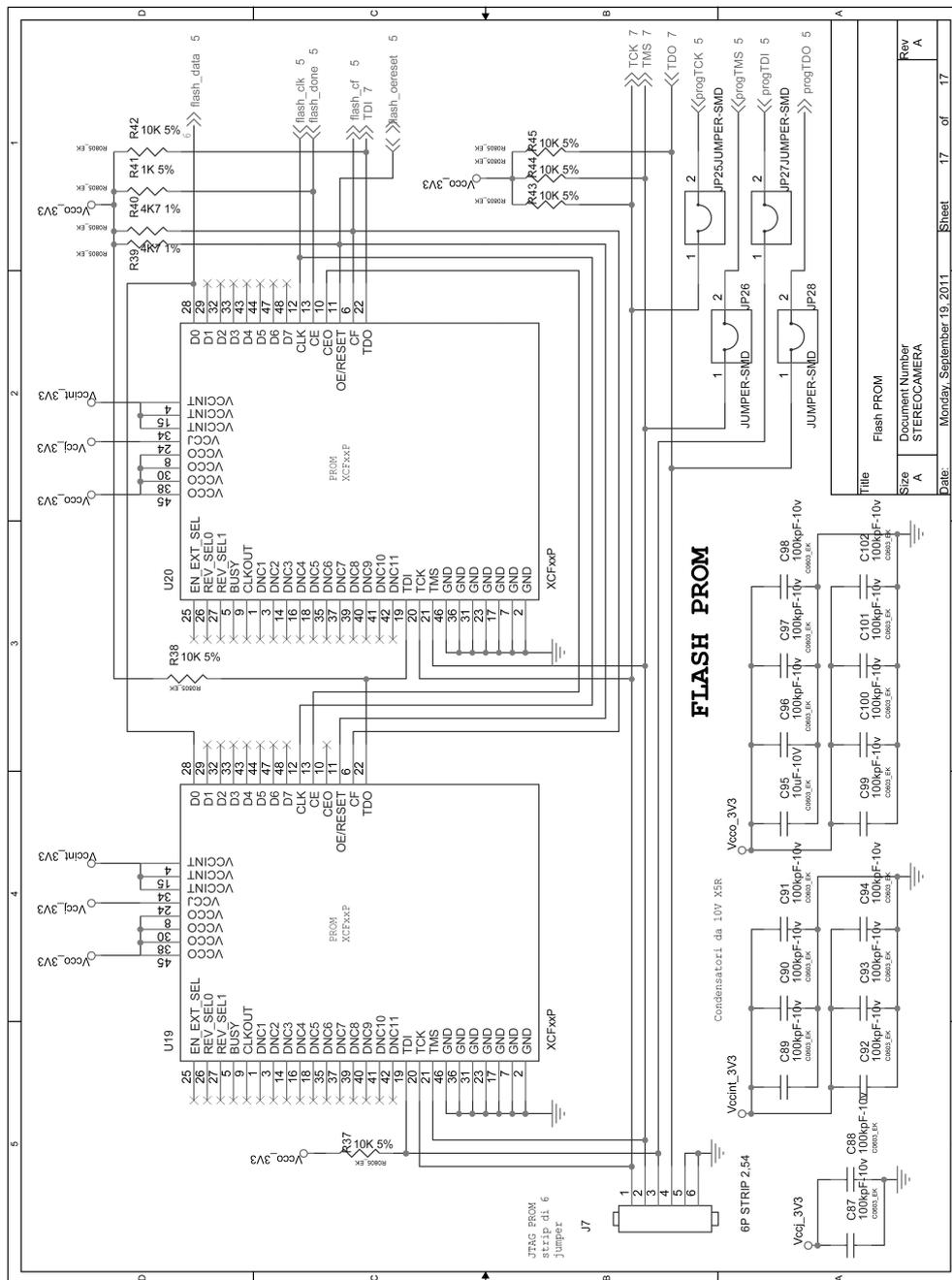
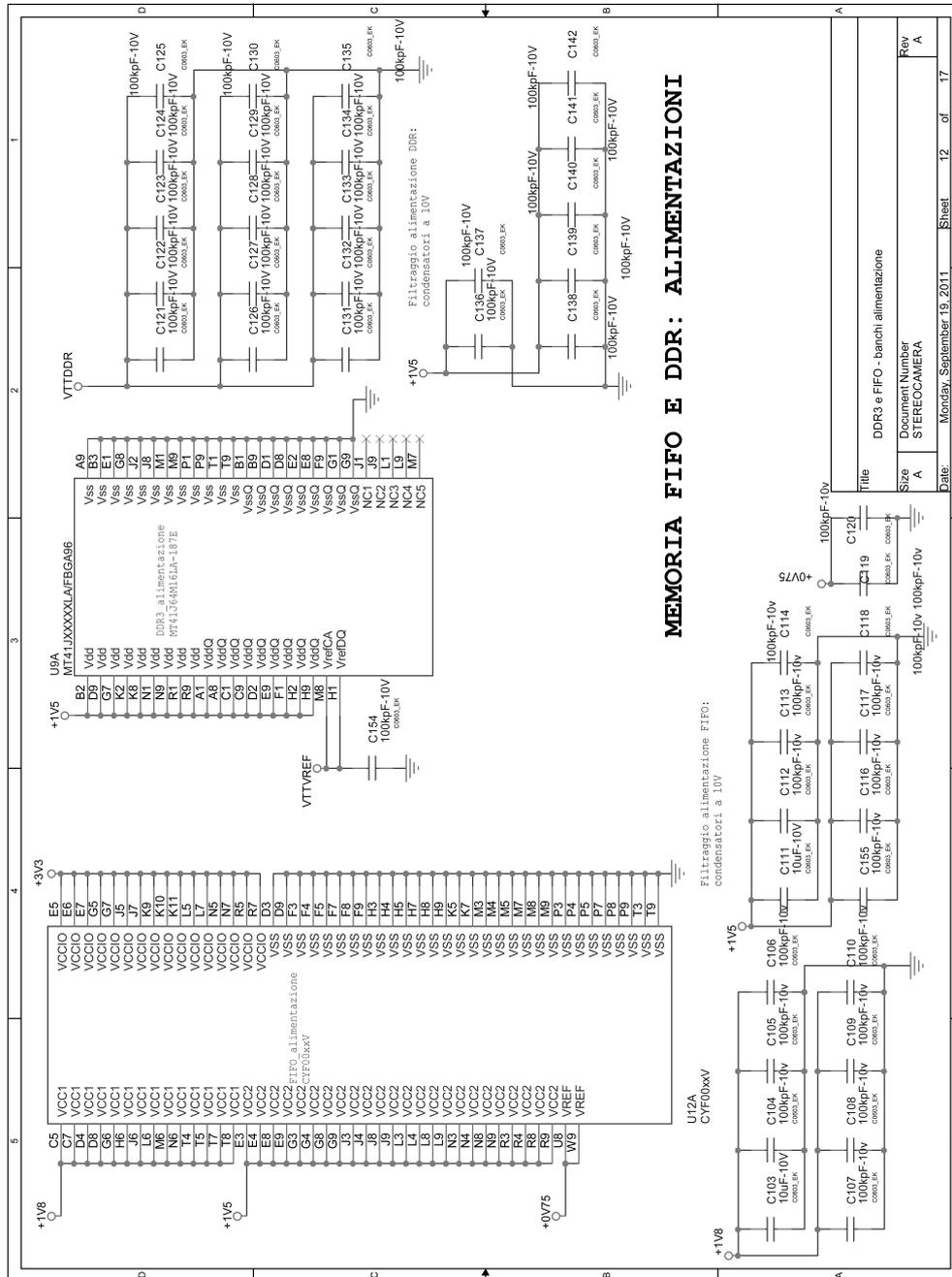


Figura A.8: FPGA: banco alimentazioni e segnali dedicati



Title		Flash PROM	
Size	Document Number	STEREOCAMERA	
Rev	Date:	Monday, September 19, 2011	Sheet 17 of 17

Figura A.10: Memorie Flash PROM



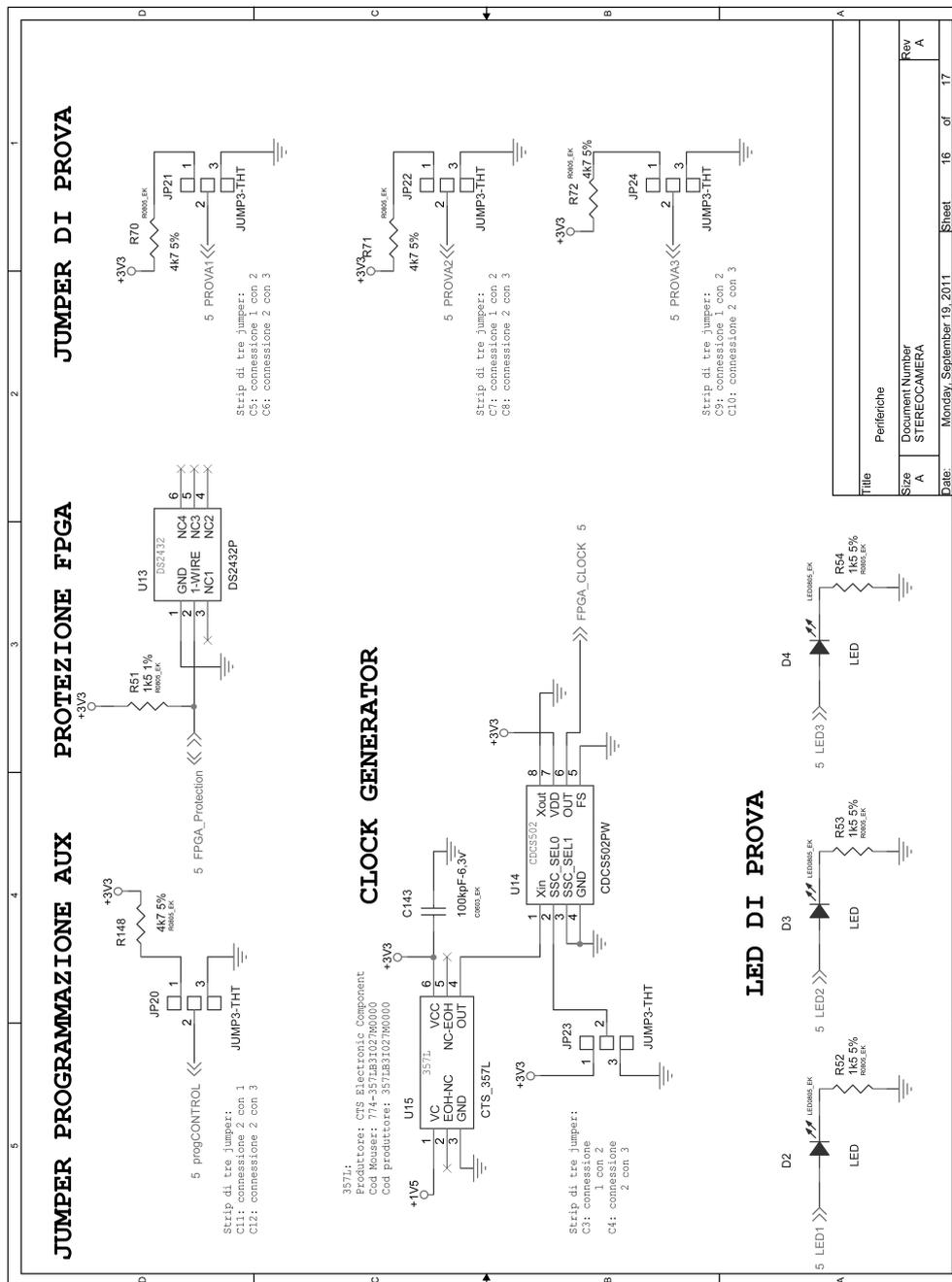


Figura A.13: Circuiti elettronici periferici

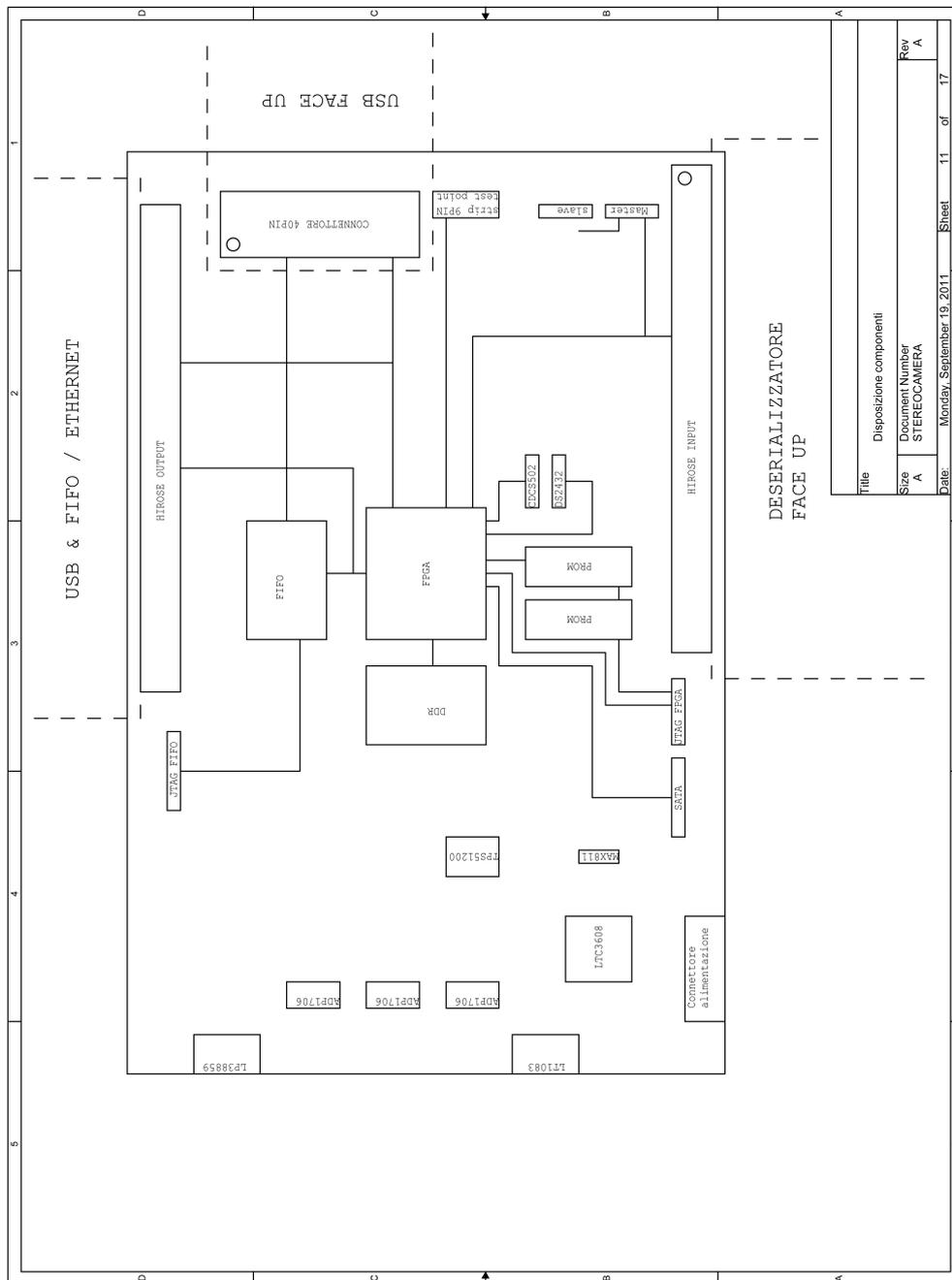


Figura A.17: Disposizione dei dispositivi

Appendice B

File UCF

```
##  
##SEGNALI DI CONTROLLO FPGA  
##  
  
NET "HSWAPEN"          LOC = "A3" ;  
NET "TCK"              LOC = "G15" ;  
NET "TDI"              LOC = "E18" ;  
NET "TMS"              LOC = "C18" ;  
NET "TDO"              LOC = "A19" ;  
NET "RFUSE"            LOC = "P15" ;  
NET "VBATT"            LOC = "R17" ;  
NET "FPGA_CLOCK"      LOC = "AA21" ;  
##NET "FPGA_CLOCK"    LOC = "Y11" ;  
NET "FPGA_Protection" LOC = "AB21" ;  
##  
##  
##SEGNALI DI DATI MEMORIA FIFO  
##
```

```

NET "FIFO_IN0"   LOC = "C5" ;
NET "FIFO_IN1"   LOC = "A5" ;
NET "FIFO_IN2"   LOC = "D6" ;
NET "FIFO_IN3"   LOC = "C6" ;
NET "FIFO_IN4"   LOC = "B6" ;
NET "FIFO_IN5"   LOC = "A6" ;
NET "FIFO_IN6"   LOC = "C7" ;
NET "FIFO_IN7"   LOC = "A7" ;
NET "FIFO_IN8"   LOC = "B8" ;
NET "FIFO_IN9"   LOC = "A8" ;
NET "FIFO_IN10"  LOC = "D9" ;
NET "FIFO_IN11"  LOC = "C8" ;
NET "FIFO_IN12"  LOC = "C9" ;
NET "FIFO_IN13"  LOC = "A9" ;
NET "FIFO_IN14"  LOC = "D7" ;
NET "FIFO_IN15"  LOC = "D8" ;
##
##
## SEGNALI DI CONTROLLO MEMORIA FIFO
##

```

```

NET "FIFO_DVAL"   LOC = "C10" ;
NET "FIFO_RCLK"   LOC = "B12" ;
NET "FIFO_WCLK"   LOC = "A12" ;
NET "FIFO_WEN"    LOC = "C14" ;
NET "FIFO_REN"    LOC = "B14" ;
NET "FIFO_LD"     LOC = "A14" ;
NET "FIFO_PAE"    LOC = "A15" ;
NET "FIFO_PAF"    LOC = "D15" ;
NET "FIFO_MRS"    LOC = "C17" ;
NET "FIFO_PRS"    LOC = "A17" ;
##
##
## SEGNALI DI CONTROLLO DESERIALIZZATORE
##

```

```

NET "inHIROSE_S_CONTROL_AD0_MASTER_1" LOC = "G16" ;
NET "inHIROSE_/RESET_1" LOC = "G17" ;
NET "inHIROSE_RPWDN_1" LOC = "F16" ;
NET "inHIROSE_SHUTDOWN/OE_1" LOC = "F17" ;
NET "inHIROSE_NEW_FPGA_OUT" LOC = "B21" ;
NET "inHIROSE_AUX_PIN_1" LOC = "B22" ;
NET "inHIROSE_AUX_PIN_2" LOC = "A20" ;
NET "inHIROSE_SDATA_1" LOC = "G22" ;
NET "inHIROSE_SCLK_1" LOC = "K19" ;
NET "inHIROSE_RCLK_1" LOC = "H21" ;
##NET "inHIROSE_RCLK_1" LOC = "K21" ;
NET "inHIROSE_LOCK_1" LOC = "M17" ;
##
##
##SEGNALI DI DATI DESERIALIZZATORE
##

```

```

NET "inRight_0" LOC = "T21" ;
NET "inRight_1" LOC = "T22" ;
NET "inRight_2" LOC = "R19" ;
NET "inRight_3" LOC = "R20" ;
NET "inRight_4" LOC = "R22" ;
NET "inRight_5" LOC = "P18" ;
NET "inRight_6" LOC = "P19" ;
NET "inRight_7" LOC = "P20" ;
NET "inLeft_0" LOC = "W20" ;
NET "inLeft_1" LOC = "W22" ;
NET "inLeft_2" LOC = "V20" ;
NET "inLeft_3" LOC = "V21" ;
NET "inLeft_4" LOC = "V22" ;
NET "inLeft_5" LOC = "U19" ;
NET "inLeft_6" LOC = "U20" ;
NET "inLeft_7" LOC = "U22" ;
##
##
##SEGNALI DI CONTROLLO USB

```

##

```
NET "USB_PA0"      LOC = "A21" ;
NET "USB_PA1"      LOC = "K16" ;
NET "USB_PA2"      LOC = "J16" ;
NET "USB_PA3"      LOC = "H16" ;
NET "USB_PA4"      LOC = "H17" ;
NET "USB_PA5"      LOC = "D19" ;
NET "USB_PA6"      LOC = "D20" ;
NET "USB_PA7"      LOC = "F18" ;
NET "USB_WAKEUP"   LOC = "F19" ;
NET "USB_/RESET"   LOC = "D21" ;
NET "USB_SDA"      LOC = "D22" ;
NET "USB_SCL"      LOC = "C20" ;
NET "USB_RDY0"     LOC = "C22" ;
NET "USB_RDY1"     LOC = "G19" ;
NET "USB_CTRL0"    LOC = "F20" ;
NET "USB_CTRL1"    LOC = "H19" ;
NET "USB_CTRL2"    LOC = "H18" ;
NET "USB_CLKOUT"   LOC = "K20" ;
##NET "USB_CLKOUT" LOC = "K18" ;
NET "USB_IFCLK"    LOC = "H22" ;
```

##

##

##SEGNALI PROM

##

```
NET "flash_done"   LOC = "Y22" ;
NET "flash_clk"    LOC = "Y21" ;
NET "flash_data"   LOC = "AA20" ;
NET "flash_oereset" LOC = "T6" ;
NET "flash_cf"     LOC = "AA1" ;
```

##

##

##SEGNALI PROGRAMMAZIONE AUSILIARIA

##

```
NET "progTCK"          LOC = "Y9" ;
NET "progTMS"          LOC = "AB9" ;
NET "progTDO"          LOC = "AA8" ;
NET "progTDI"          LOC = "AB8" ;
NET "progCONTROL"     LOC = "AA10" ;
##
##
##SEGNALI SATA
##

NET "CLKP_SATA"        LOC = "W12" ;
NET "CLKN_SATA"        LOC = "Y12" ;
##NET "CLKN_SATA"      LOC = "Y15" ;
##NET "CLKP_SATA"      LOC = "AB15" ;
NET "LVDS_DATAP_SATA"  LOC = "V11" ;
NET "LVDS_DATAN_SATA"  LOC = "W11" ;
##
##
##SEGNALI DI DATI MEMORIA DDR
##
```

```
NET "DDR_DQ0" LOC = "N3" ;
NET "DDR_DQ1" LOC = "N1" ;
NET "DDR_DQ2" LOC = "M2" ;
NET "DDR_DQ3" LOC = "M1" ;
NET "DDR_DQ4" LOC = "J3" ;
NET "DDR_DQ5" LOC = "J1" ;
NET "DDR_DQ6" LOC = "K2" ;
NET "DDR_DQ7" LOC = "K1" ;
NET "DDR_DQ8" LOC = "P2" ;
NET "DDR_DQ9" LOC = "P1" ;
NET "DDR_DQ10" LOC = "R3" ;
NET "DDR_DQ11" LOC = "R1" ;
NET "DDR_DQ12" LOC = "U3" ;
NET "DDR_DQ13" LOC = "U1" ;
NET "DDR_DQ14" LOC = "V2" ;
NET "DDR_DQ15" LOC = "V1" ;
##
##
##SEGNALI DI INDIRIZZI MEMORIA DDR
##
```

```
NET "DDR_A0"    LOC = "H2" ;
NET "DDR_A1"    LOC = "H1" ;
NET "DDR_A2"    LOC = "H5" ;
NET "DDR_A3"    LOC = "K6" ;
NET "DDR_A4"    LOC = "F3" ;
NET "DDR_A5"    LOC = "K3" ;
NET "DDR_A6"    LOC = "J4" ;
NET "DDR_A7"    LOC = "H6" ;
NET "DDR_A8"    LOC = "E3" ;
NET "DDR_A9"    LOC = "E1" ;
NET "DDR_A10"   LOC = "G4" ;
NET "DDR_A11"   LOC = "C1" ;
NET "DDR_A12"   LOC = "D1" ;
NET "DDR_A13"   LOC = "G6" ;
NET "DDR_A14"   LOC = "F5" ;
NET "DDR_BA0"   LOC = "G3" ;
NET "DDR_BA1"   LOC = "G1" ;
NET "DDR_BA2"   LOC = "F1" ;
##
##
##SEGNALI DI CONTROLLO MEMORIA DDR
##
```

```
NET "DDR_UDQS"      LOC = "T2" ;
NET "DDR_UDQS#"    LOC = "T1" ;
NET "DDR_LDQS"     LOC = "L3" ;
NET "DDR_LDQS#"    LOC = "L1" ;
NET "DDR_UDM"      LOC = "M3" ;
NET "DDR_LDM"      LOC = "L4" ;
NET "DDR_RAS"      LOC = "K5" ;
NET "DDR_CAS"      LOC = "K4" ;
NET "DDR_ODT"      LOC = "J6" ;
NET "DDR_CKP"      LOC = "H4" ;
NET "DDR_CKN"      LOC = "H3" ;
NET "DDR_CKE"      LOC = "D2" ;
NET "DDR_WE"       LOC = "F2" ;
NET "DDR_RESET"    LOC = "C3" ;
```

```
##
```

```
##
```

```
##SEGNALI DI TEST
```

```
##
```

```
NET "T1"           LOC = "E20" ;
NET "T2"           LOC = "E22" ;
NET "T3"           LOC = "J17" ;
NET "T4"           LOC = "K17" ;
NET "T5"           LOC = "F21" ;
NET "T6"           LOC = "F22" ;
NET "T7"           LOC = "H20" ;
NET "T8"           LOC = "J19" ;
NET "T9"           LOC = "G20" ;
NET "PROVA1"       LOC = "Y19" ;
NET "PROVA2"       LOC = "AB19" ;
NET "PROVA3"       LOC = "Y17" ;
NET "LED1"         LOC = "AB17" ;
NET "LED2"         LOC = "AA16" ;
NET "LED3"         LOC = "AB16" ;
```

Appendice C

Layout

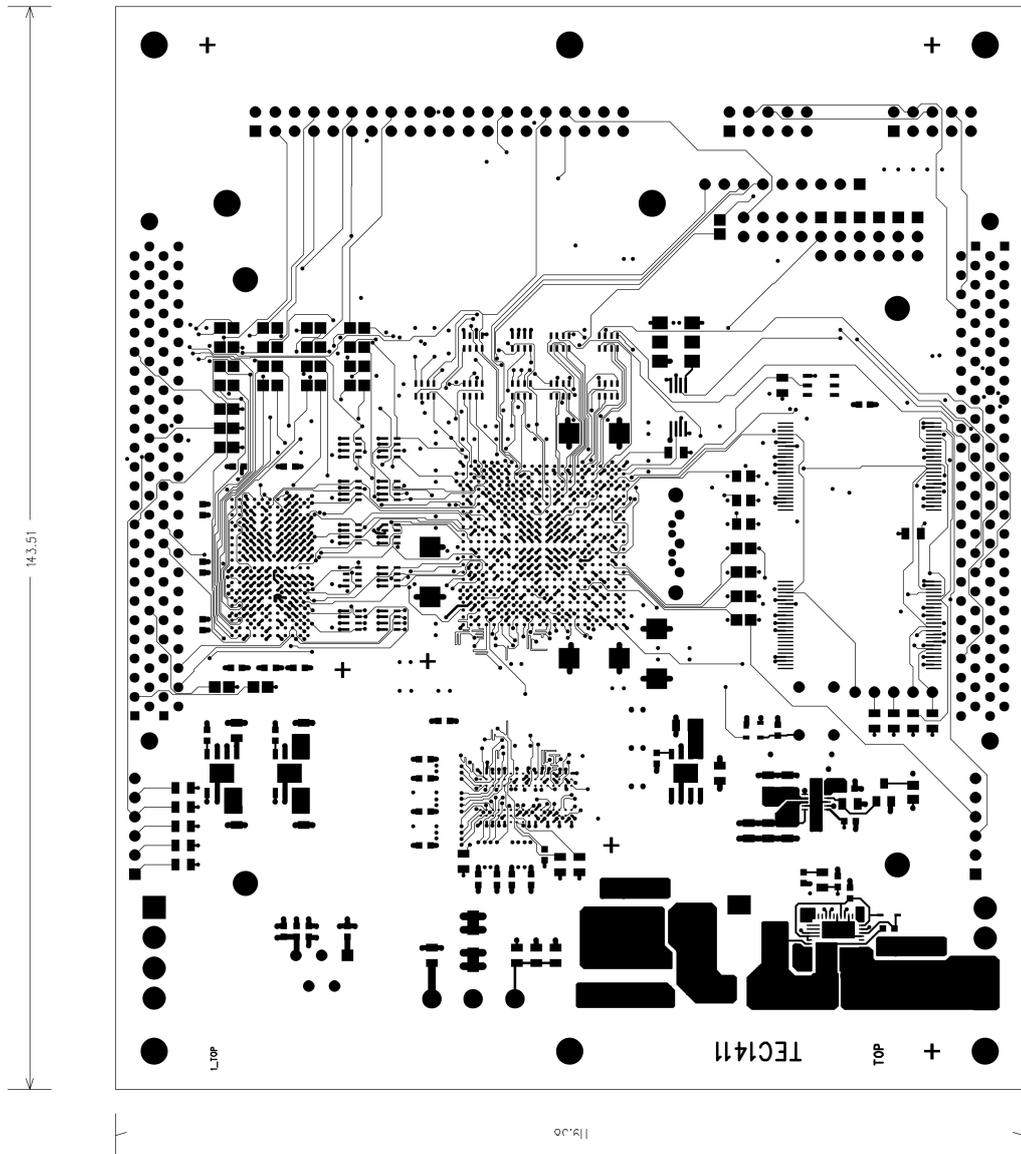


Figura C.1: Lato componenti

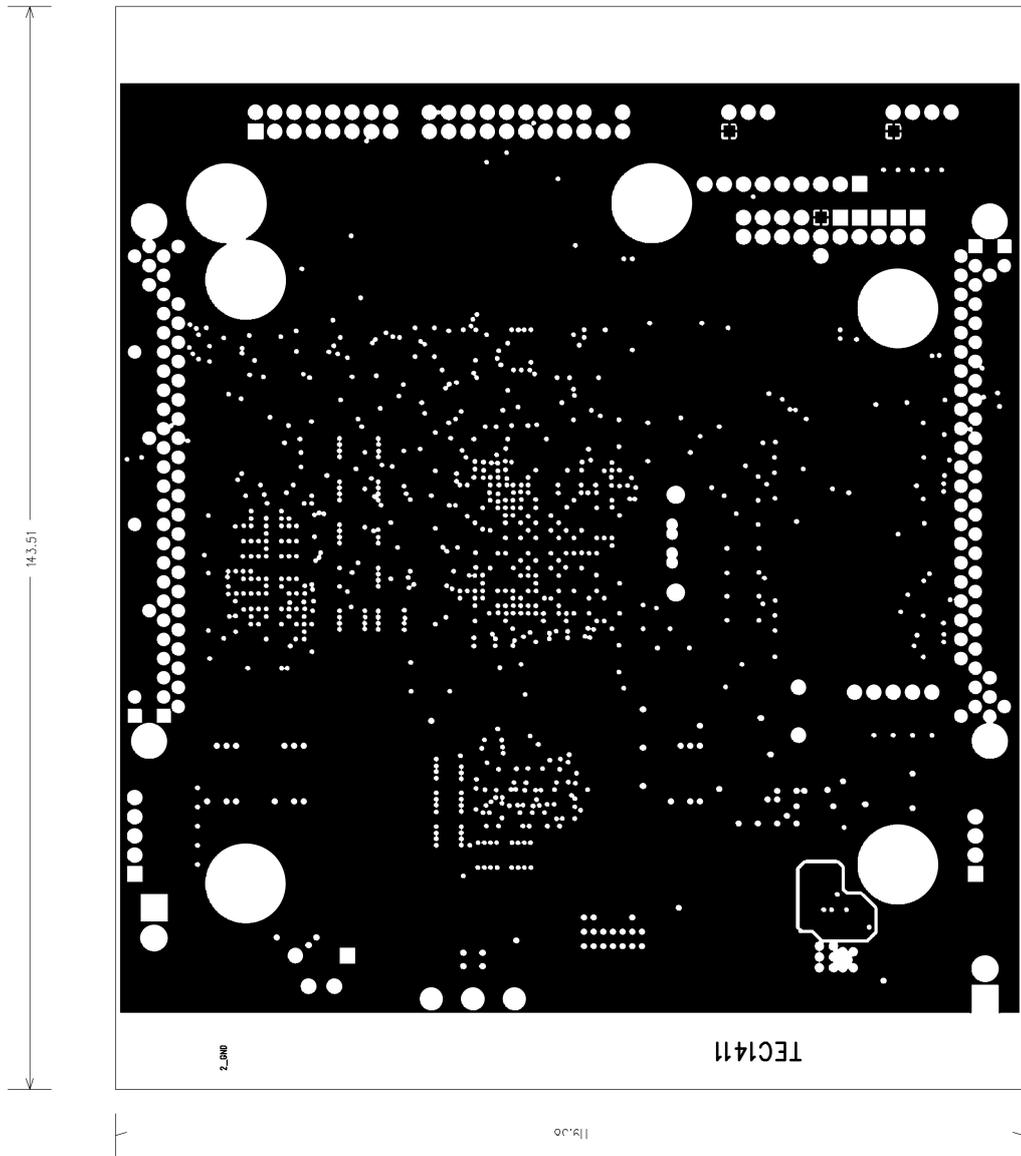


Figura C.2: Piano interno di massa

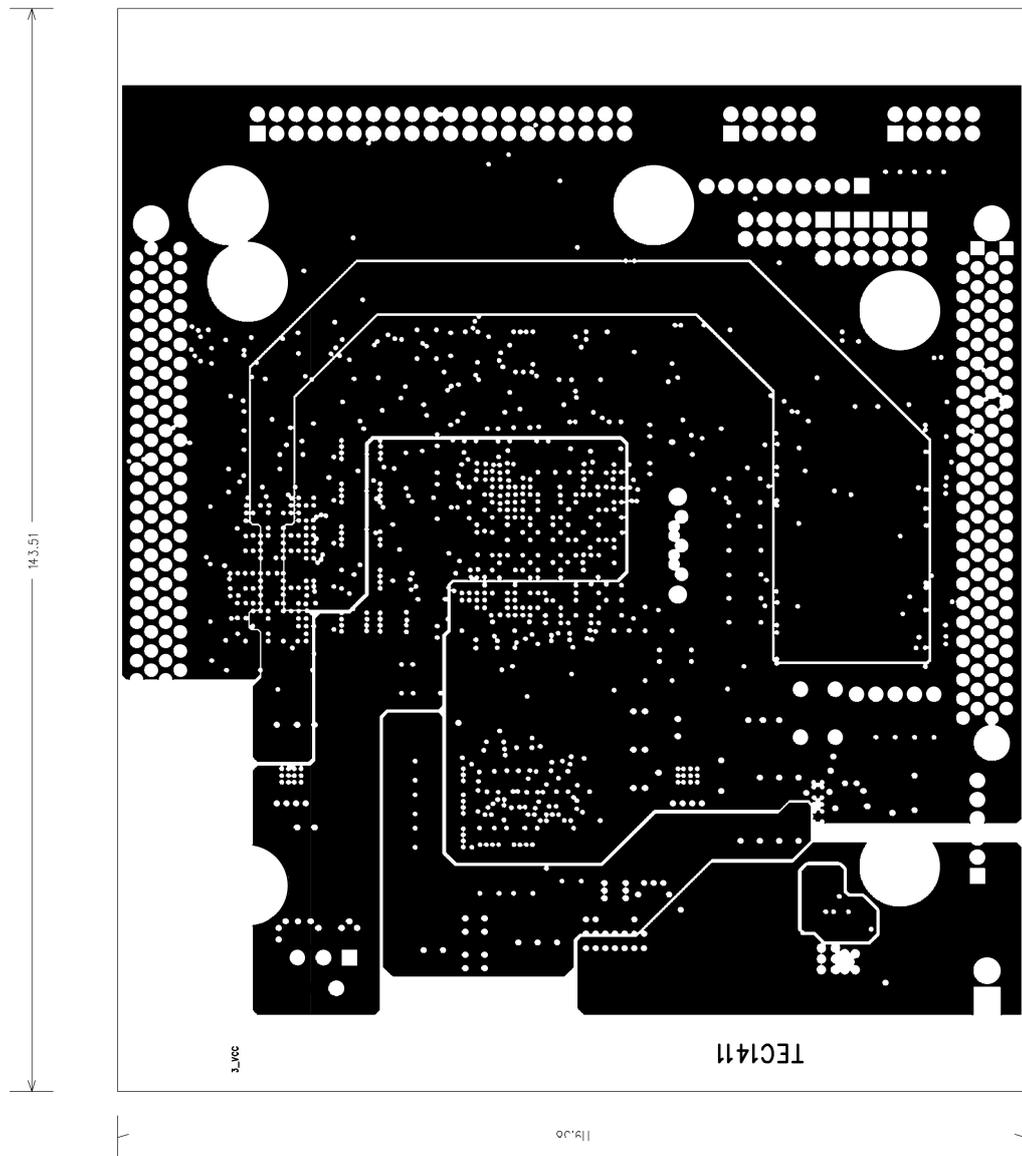


Figura C.3: Piano interno alimentazioni

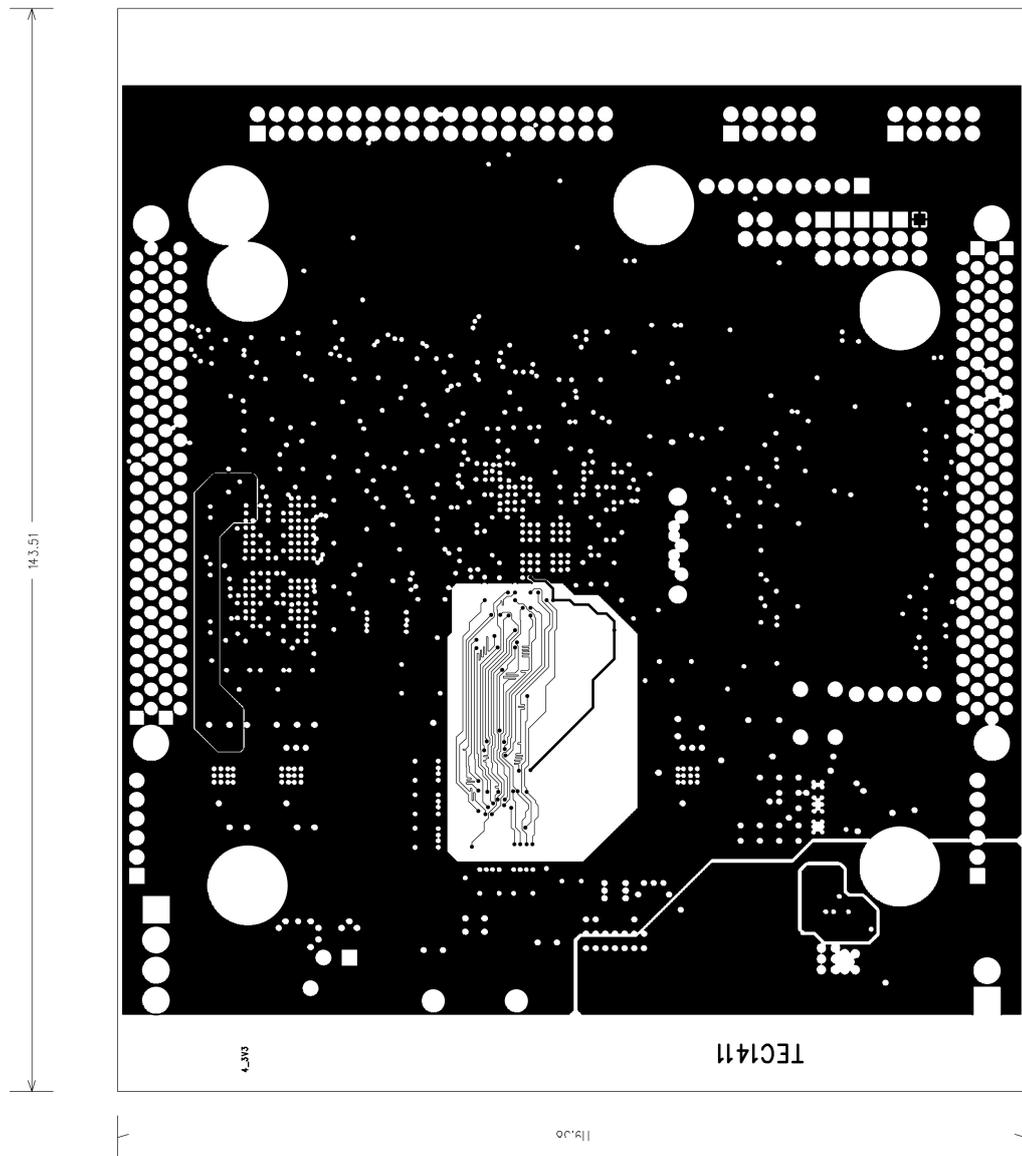


Figura C.4: Piano interno alimentazioni e segnali

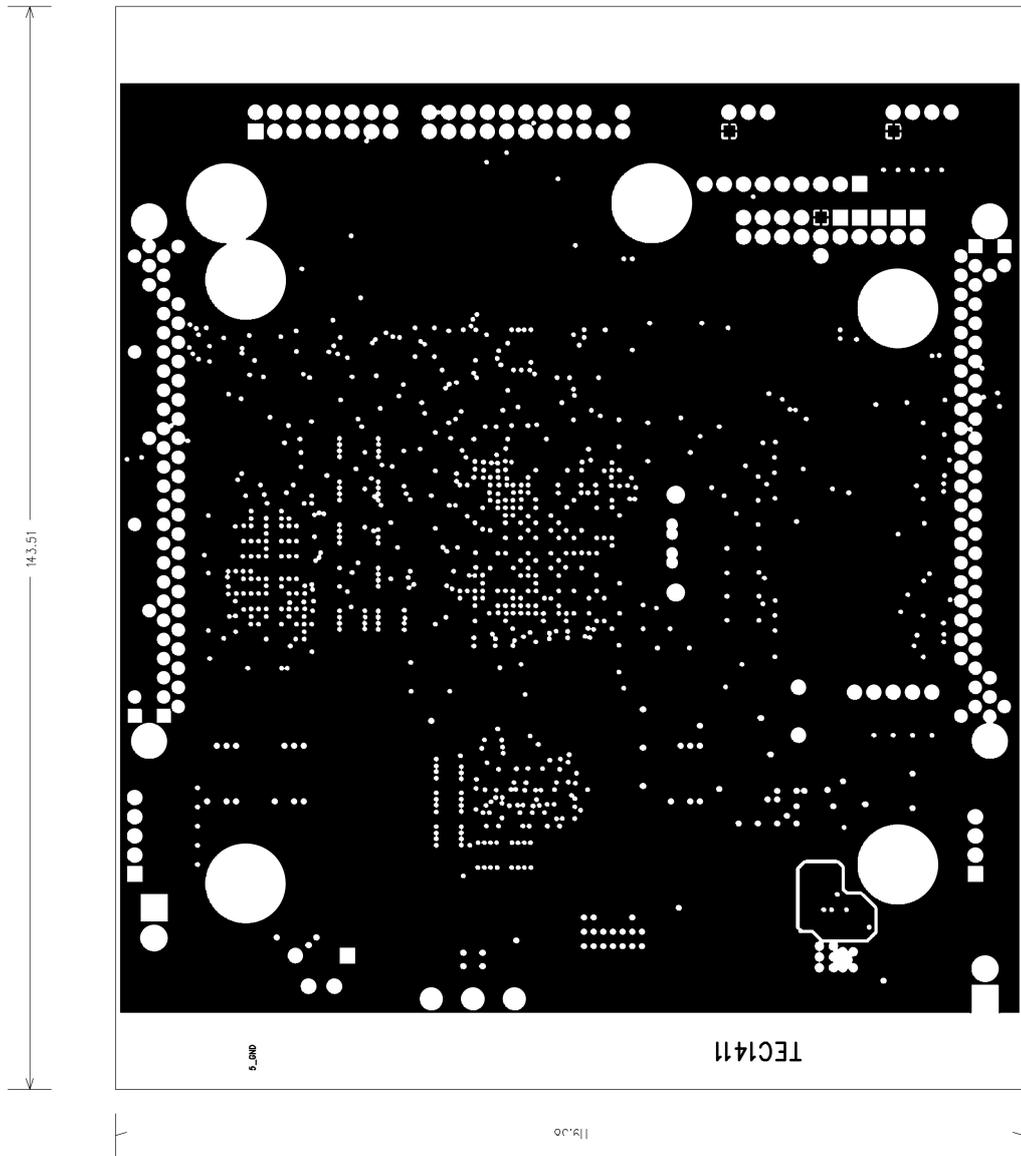


Figura C.5: Piano interno massa

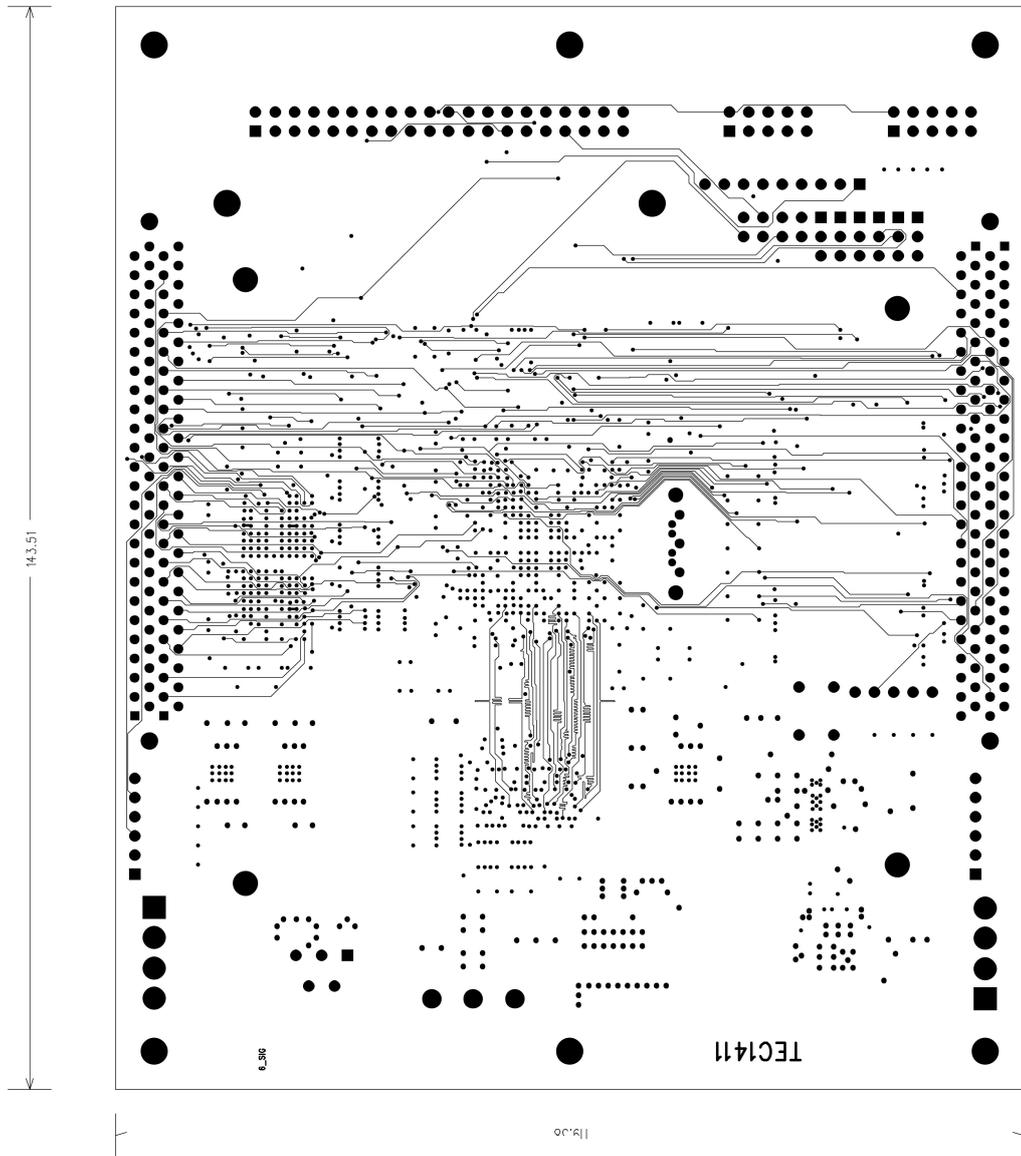


Figura C.6: Piano interno segnali

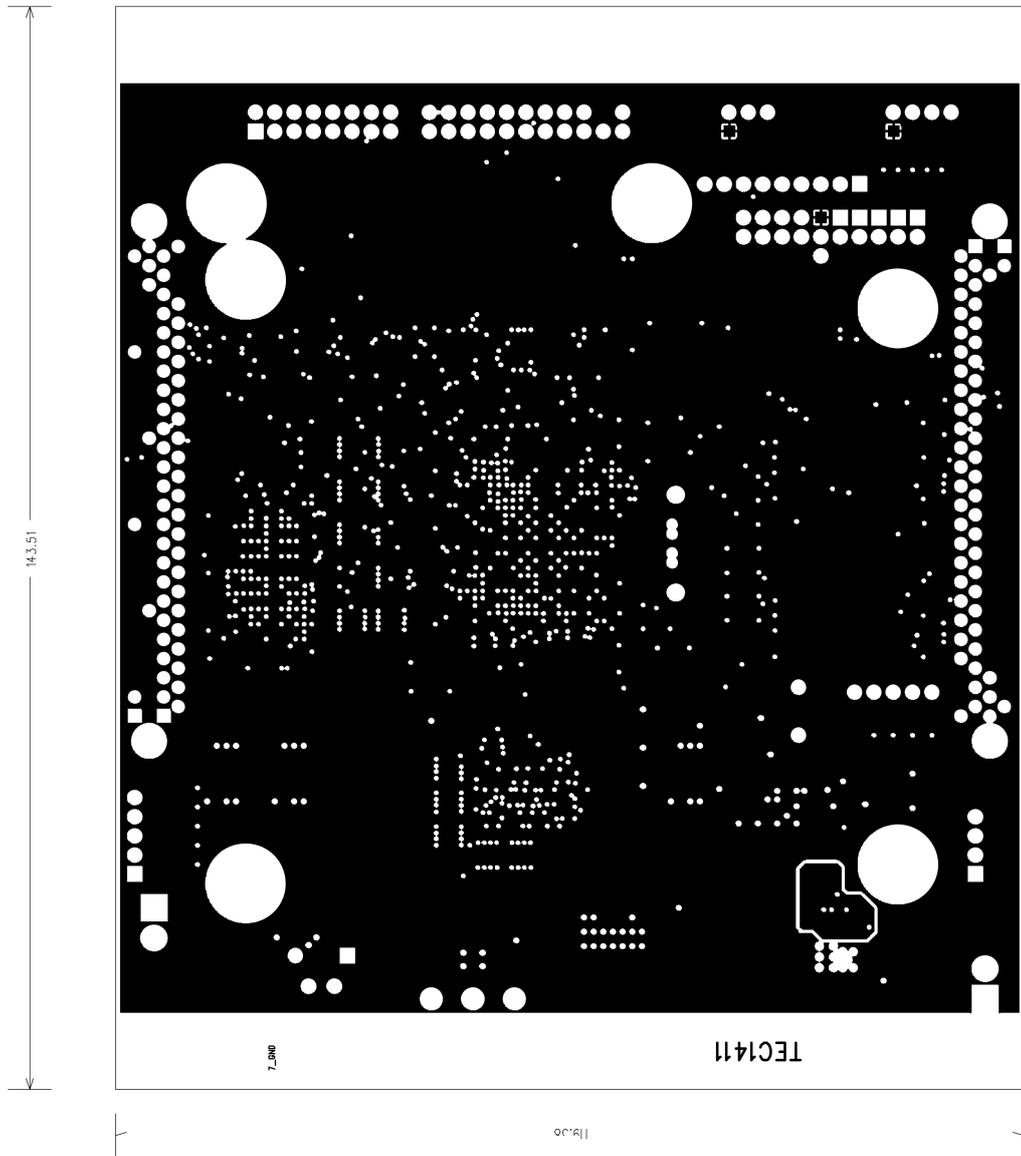


Figura C.7: Piano interno massa

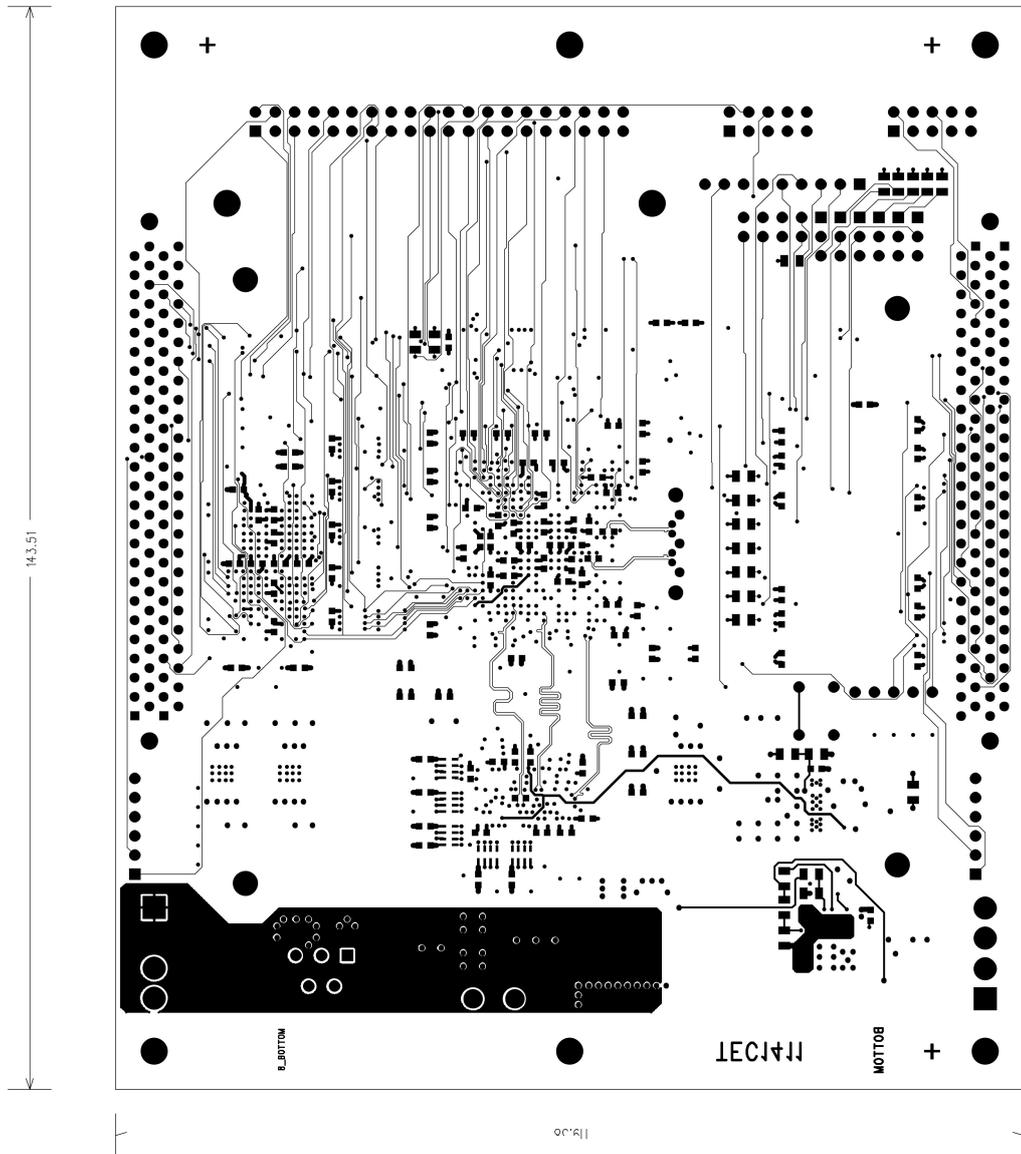


Figura C.8: Lato saldature

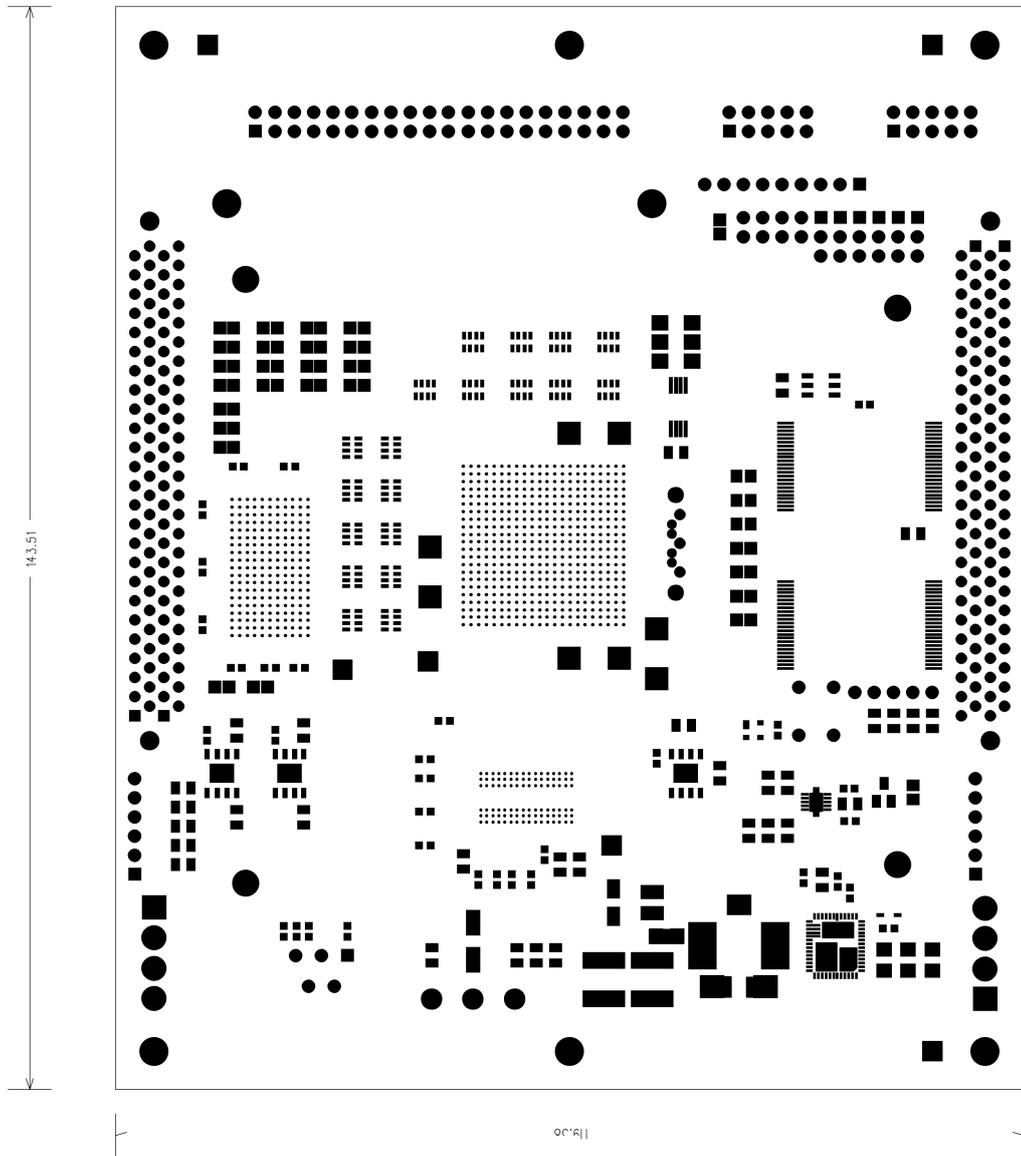


Figura C.9: Solder lato componenti

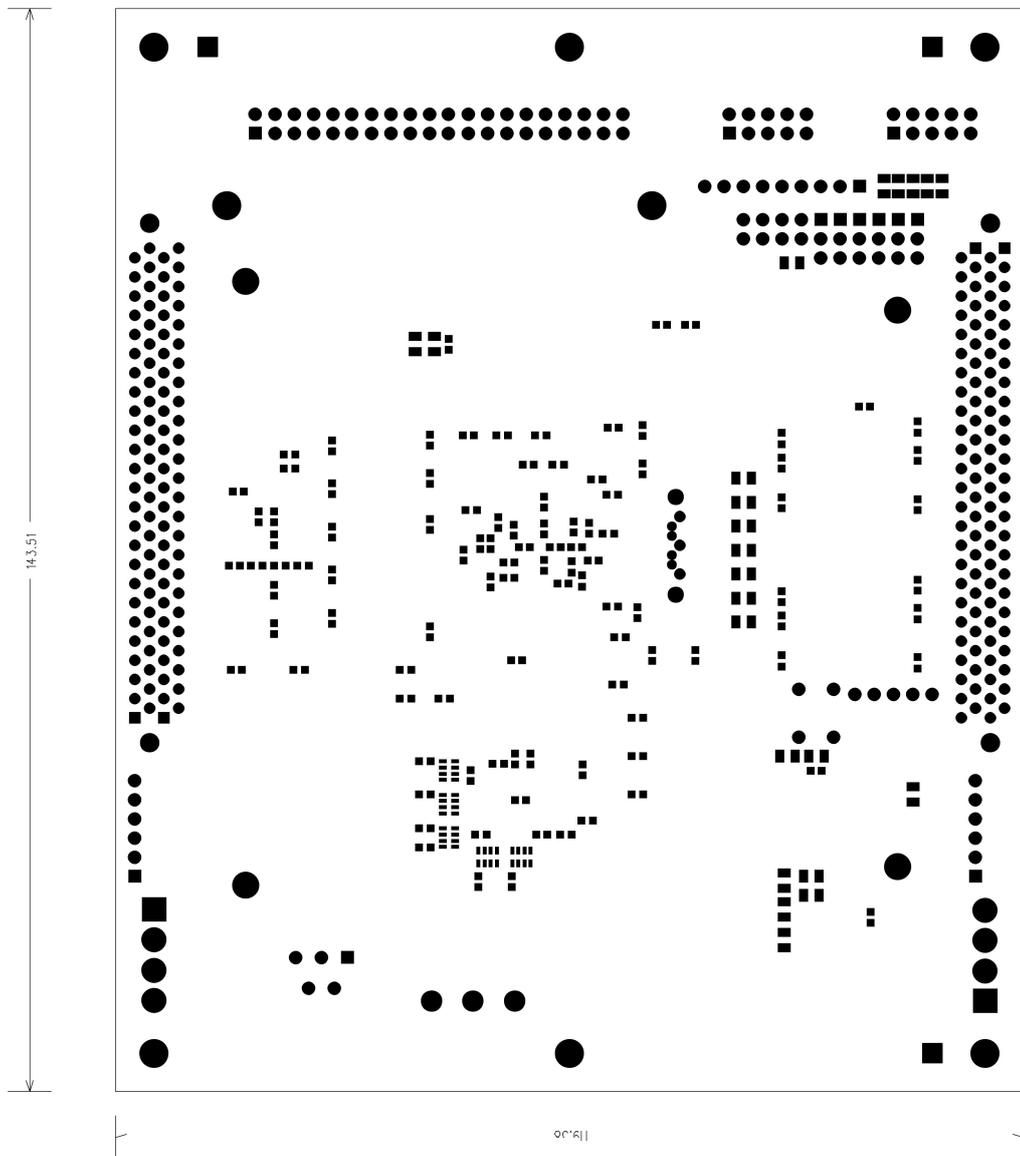


Figura C.10: Solder lato saldature

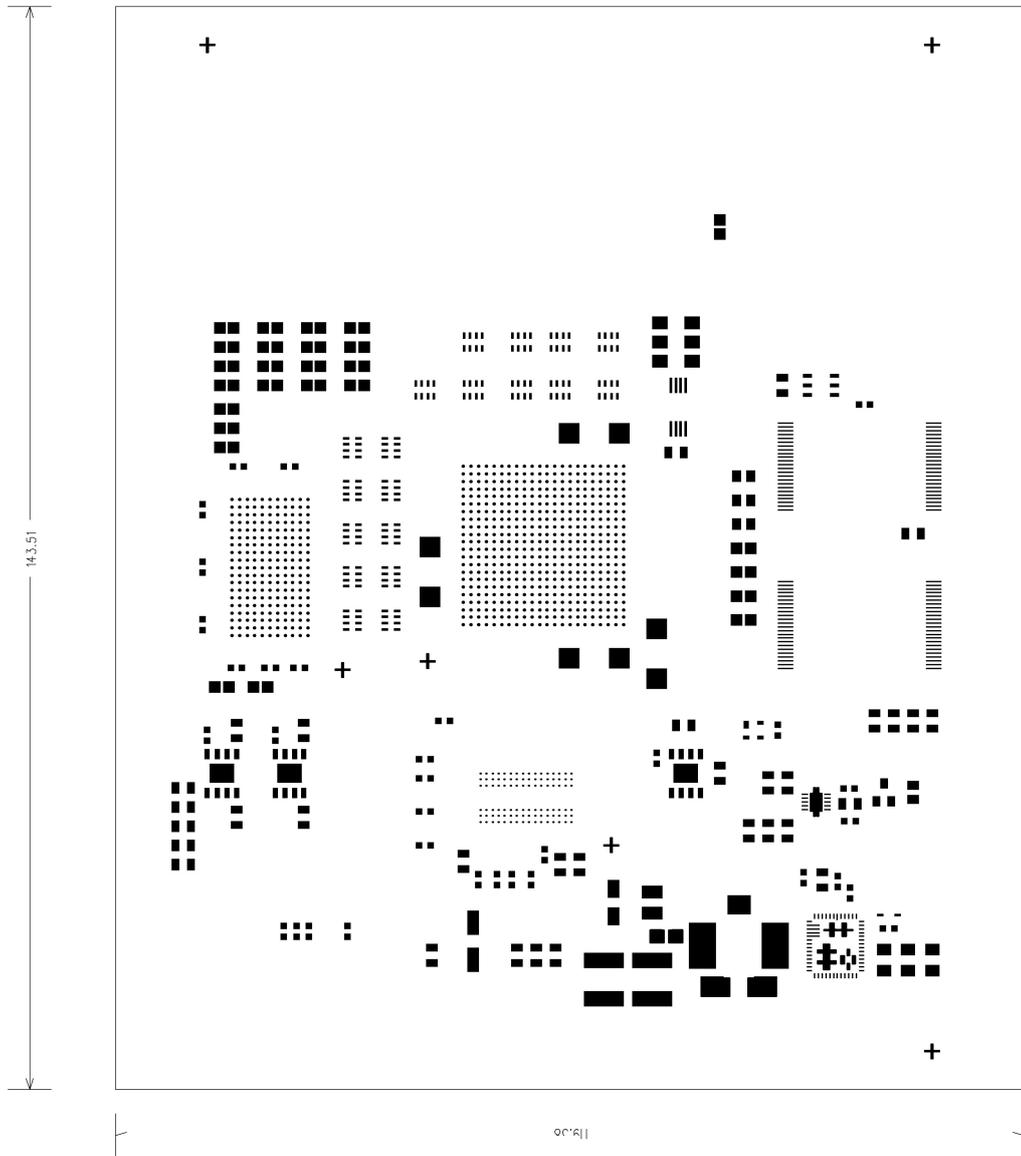


Figura C.11: Pasta salda SMD lato componenti

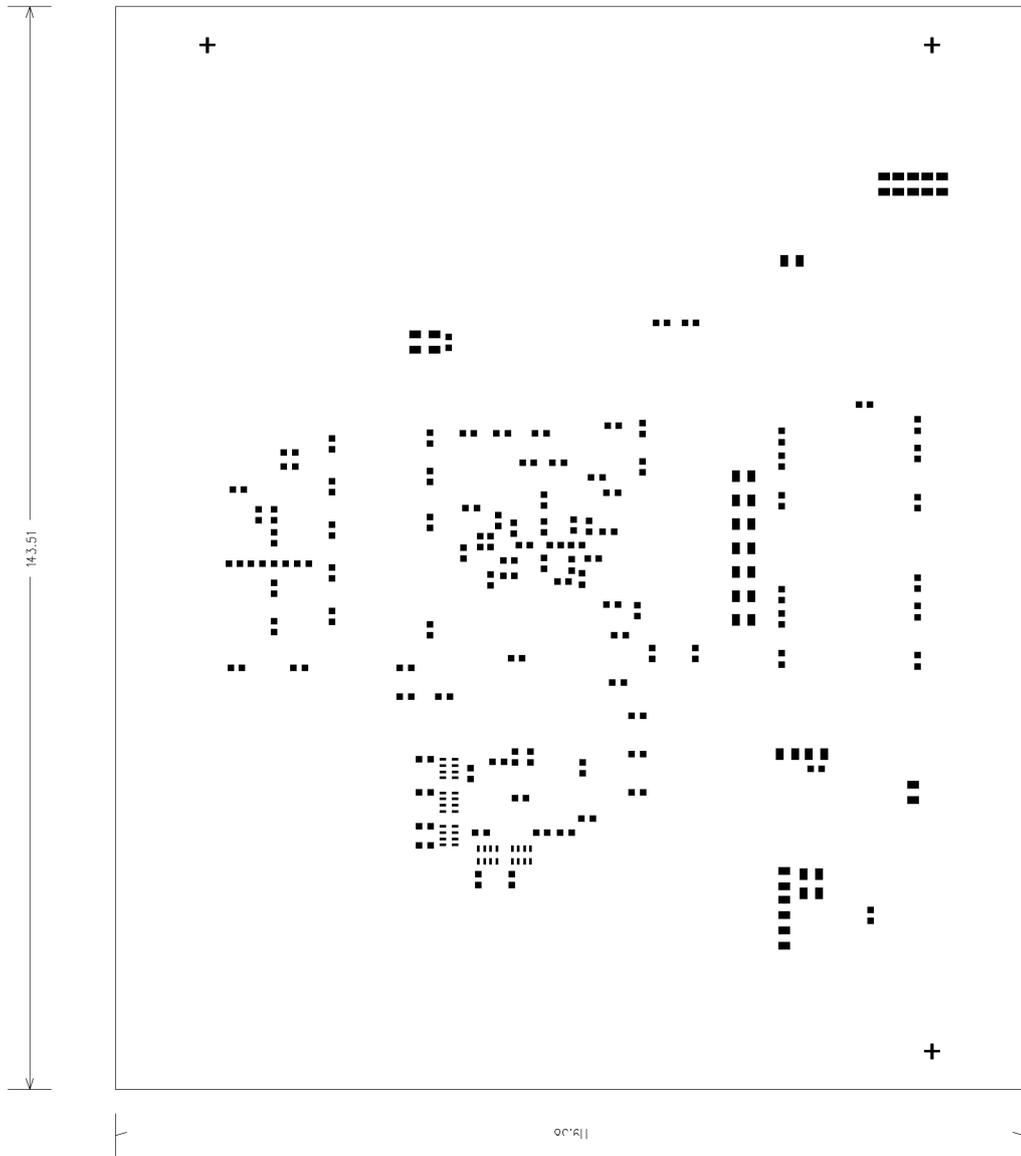
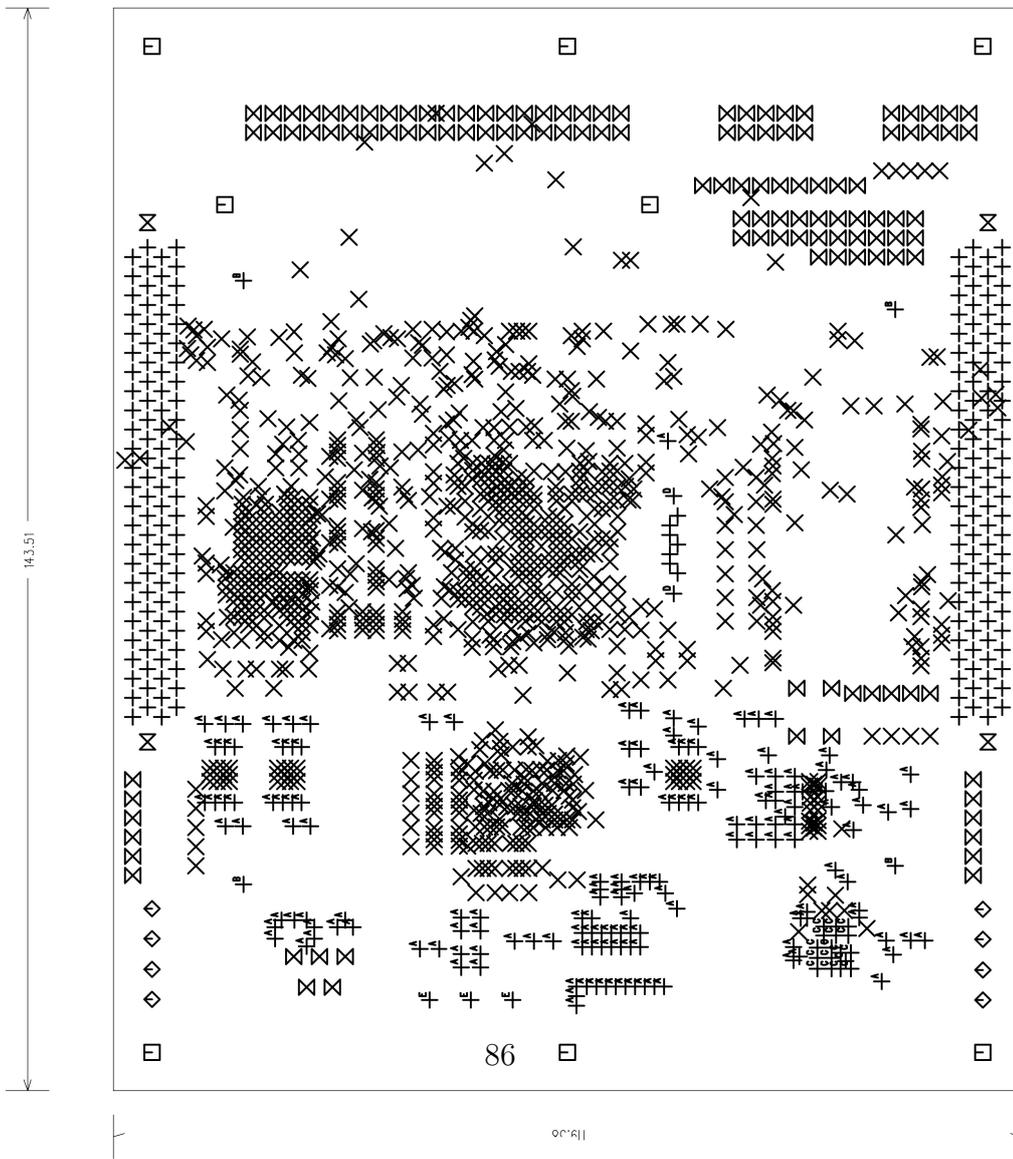


Figura C.12: Pasta salda SMD lato saldature



SIZE	QTY	SYM	PL
0.762	207	+	YE
0.254	1067	X	YE
3.556	8	□	NC
2.032	8	◇	YE
1.016	121	⊗	YE
2.286	4	⊗	NC
0.3556	160	⊕ ^A	YE
3.302	4	⊕ ^B	NC
0.4064	17	⊕ ^C	YE
1.397	2	⊕ ^D	YE
1.778	3	⊕ ^E	YE

Figura C.13: Specifiche di foratura

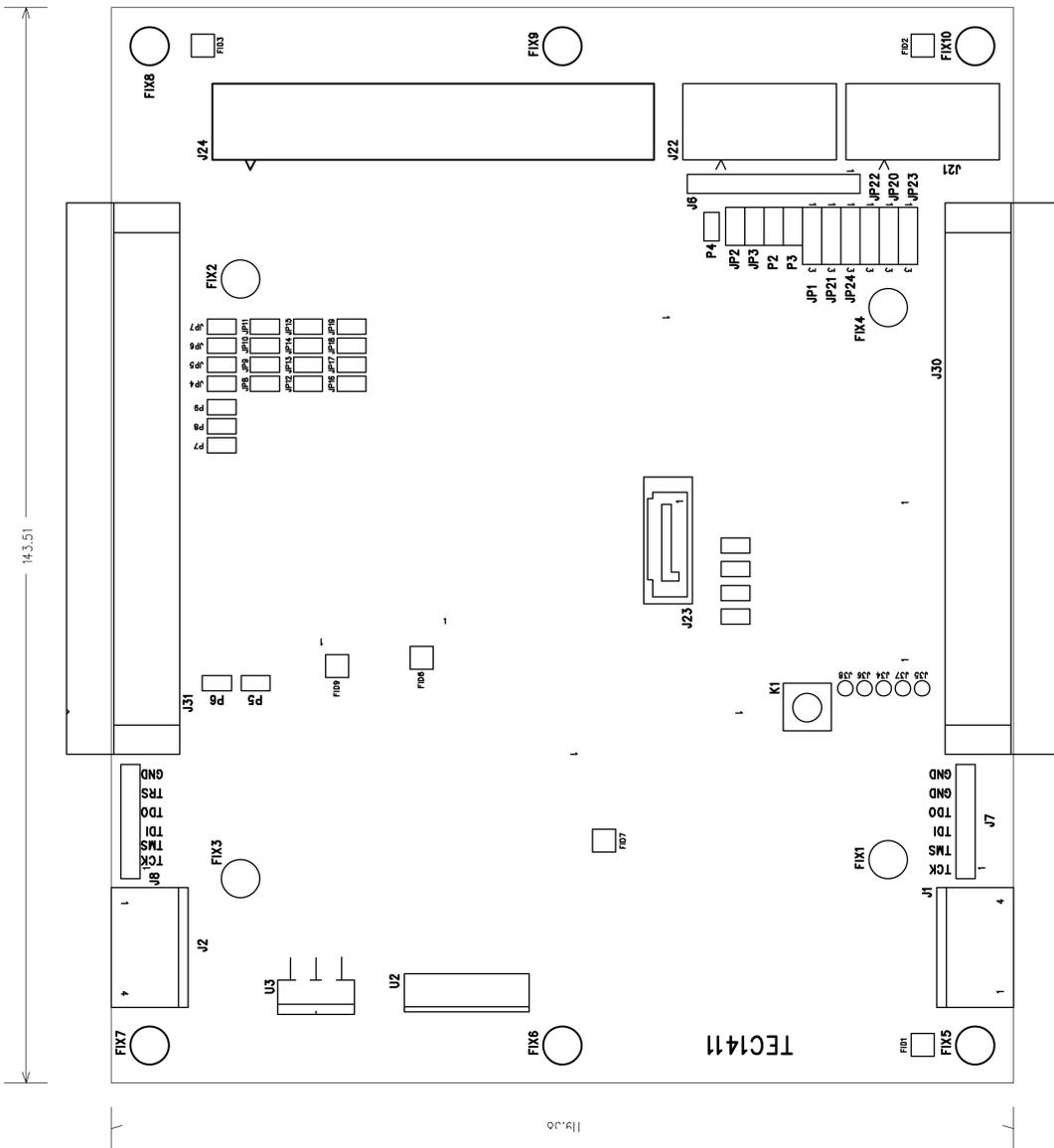


Figura C.14: Serigrafia lato componenti (solo componenti through-hole)

Bibliografia

- [1] Stefano Mattocchia, *“Introduzione alla Visione Stereo”*, Novembre 2008.
- [2] Xilinx inc., *“Spartan-6 FPGA Configuration User Guide, UG380 (v2.2)”*, 30 Luglio 2010.
- [3] Xilinx inc., *“Spartan-6 FPGA SelectIO Resources, UG381 (v1.4)”*, 16 Dicembre 2010.
- [4] Xilinx inc., *“SP605 Hardware User Guide, UG526 (v1.5)”*, 16 Febbraio 2011.
- [5] Xilinx inc., *“Spartan-6 Family Overview, DS160 (v1.3)”*, 5 Novembre 2009.
- [6] Xilinx inc., *“Spartan-6 FPGA Data Sheet: DC and Switching Characteristics, DS162 (v1.12)”*, 11 Febbraio 2011.
- [7] Xilinx inc., *“Spartan-6 FPGA Packaging and Pinouts, UG385 (v2.0)”*, 24 Febbraio 2011.
- [8] Xilinx inc., *“Platform Flash PROM User Guide, UG161 (v1.5)”*, 26 Ottobre 2009.
- [9] Xilinx inc., *“Spartan-6 FPGA Clocking Resources, UG382 (v1.5)”*, 16 Febbraio 2011.
- [10] Xilinx inc., *“Spartan-6 FPGA Memory Controller, UG388 (v2.3)”*, 9 Agosto 2010.
- [11] Xilinx inc., *“XAPP694 (v1.1.1)”*, 19 Novembre 2007.
- [12] Xilinx inc., *“XAPP780 (v1.1)”*, 28 Maggio 2010.

- [13] Xilinx inc., “*SP605 Schematics, XTP067*”.
- [14] Xilinx inc., “*Platform Flash In-System Programmable Configuration PROMS, DS123 (v2.6)*”, 14 Marzo 2005.
- [15] CTS Electronic Component inc., “*Model 357, Voltage Controlled Crystal Oscillator (rev H)*”.
- [16] Analog Devices inc., “*ADP1706/ADP1707/ADP1708, 1 A, Low Dropout, CMOS Linear Regulator (rev 0)*”.
- [17] Texas Instruments inc., “*CDCS502, Crystal Oscillator / Clock Generator with optional SSC*”, Dicembre 2008.
- [18] Cypress Semiconductor Corp., “*CYF0018V/CYF0036V/CYF0072V, 18/36/72 Mbit Programmable FIFOs (rev H)*”, 31 Marzo 2011.
- [19] Micron Technology inc., “*MT41J256M4/MT41J128M8/MT41J64M16, 1Gb: x4, x8, x16 DDR3 SDRAM (rev K)*”, Agosto 2011.
- [20] National Semiconductor Corp., “*LP38859, 3A Fast-Response High-Accuracy LDO Linear Regulator with Soft-Start (rev D)*”, Novembre 2006.
- [21] Maxim Integrated Products inc., “*DS2432, 1Kb Protected 1-Wire EEPROM with SHA-1 Engine*”, Febbraio 2010.
- [22] Linear Technology Corp., “*LT1083/LT1084/LT1085, 7.5A, 5A, 3A Low Dropout Positive Adjustable Regulators (rev G)*”, Aprile 2009.
- [23] Linear Technology Corp., “*LTC3608, 18V, 8A Monolithic Synchronous Step-Down DC/DC Converter (rev C)*”, Giugno 2010.
- [24] Maxim Integrated Products inc., “*MAX811/MAX812, 4-Pin P Voltage Monitors with Manual Reset Input (rev 4)*”, Dicembre 2005.
- [25] Silicon Laboratories inc., “*Si500D Differential Output Silicon Oscillator (rev 1.0)*”, Maggio 2011.
- [26] Texas Instruments inc., “*TPS51200, Sink/Source DDR Termination Regulator*”, Febbraio 2008.

[27] Cadence Design Systems inc., “*OrCAD Capture User Guide (v16.2)*”,
Novembre 2008.

Ringraziamenti

Giunto al termine del percorso di laurea specialistica, è doveroso ringraziare tutti coloro che mi hanno supportato ed aiutato a raggiungere questo traguardo in particolare i miei genitori per avermi dato la possibilità di studiare Ingegneria Elettronica a Bologna. Grazie anche a tutti i genitori, parenti, e amici che mi hanno sostenuto durante questi anni di studi.

Ringrazio vivamente l'Ing. Stefano Mattocchia, il Prof. Giovanni Neri, l'Ing. Michele Borgatti e l'Ing. Davide Nanni per la loro grande disponibilità e competenza.

Grazie a Derek per avermi supportato e per aver condiviso questi anni di Università.

Ancora un grazie ad Anna per le numerose correzioni della tesi.