

UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
LAUREA SPECIALISTICA IN INGEGNERIA ELETTRONICA

PROGETTAZIONE HARDWARE
DI UN SISTEMA A MICROPROCESSORE
BASATO SU CORTEX-A8 E UN DISPLAY GRAFICO
PER LA GESTIONE DI UNA INTERFACCIA
UOMO-MACCHINA

LAUREANDO: *Matteo Smania*

RELATORE: Prof. Andrea Neviani

CORRELATORE: Ing. Andrea Tellatin

Padova, 15 Marzo 2011

*Ai miei genitori,
con affetto.*

Indice

Introduzione	2
1 Panoramica sui sistemi embedded	3
1.1 Esempi di sistemi embedded	4
1.2 La piattaforma	9
1.3 Le interfacce utente	10
1.4 Sistema operativo	12
2 Progettazione del sistema	13
2.1 Definizione delle specifiche di progetto	14
2.2 Schema a blocchi del sistema	17
2.3 Scelta dell'architettura e setup della piattaforma software	18
2.3.1 La piattaforma Pandaboard OMAP4430	19
2.3.2 La piattaforma Freescale® i.MX53	21
2.3.3 La piattaforma Texas Instruments® Sitara	25
2.4 Studio del consumo di potenza	32
2.4.1 I domini di tensione del sistema	32
2.4.2 Gestione del risparmio energetico per le memorie	34
2.5 La sequenza di avvio del sistema	36
2.6 I clock di sistema	37
2.7 I bus e le periferiche di sistema	40
2.7.1 I Timer	40
2.7.2 Il bus di comunicazione I ² C	42
2.7.3 Il bus USB	43
2.7.4 Il sistema audio	47
2.7.5 Lo slot di espansione SecureDigital	51
2.7.6 La porta RS-232	54

2.7.7	Il sistema video	56
3	Dimensionamento dello stadio di alimentazione	63
3.1	I convertitori a commutazione (SMPS)	63
3.1.1	Il convertitore abbassatore di tensione	65
3.1.2	I regolatori di tensione lineari (LDO)	68
3.2	Dimensionamento dei domini di tensione	75
3.2.1	Il powermanager	75
3.2.2	Generazione della tensione 5V	76
4	Schematici di progetto	81
4.1	Schematici	81
5	Considerazioni sul layout e sulle emissioni elettromagnetiche	103
5.1	Tecniche di layout conformi alle normative	104
5.1.1	Il riferimento dei potenziali	104
5.1.2	Influenza della non idealità della massa	106
5.1.3	Il piano di massa	110
5.1.4	Impiego di 4 o 6 strati nel PCB	112
5.1.5	Tecniche di protezione da sovratensioni e sovracorrenti	116
	Conclusioni	120
	Ringraziamenti	122
	Elenco delle figure	124
	Bibliografia	125

Introduzione

L'idea che sta alla base di questo lavoro di tesi è quella di produrre l'hardware su quale è basata un *HMI*, ovvero un dispositivo di interfaccia uomo—macchina: essendo un tipo di dispositivo elettronico che interagisce *direttamente* con l'uomo, è soggetto ad un'analisi (e, di conseguenza, ad una progettazione) a più livelli (il livello ergonomico o quello prestazionale, ad esempio).

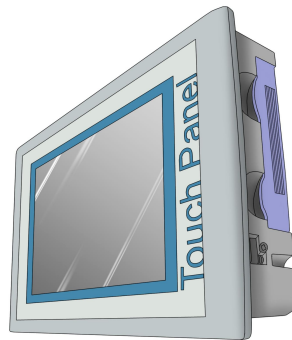


Figura 1: Un esempio di HMI

In generale, lo sforzo di ingegnerizzazione di un'interfaccia uomo—macchina è quello di produrre un'interfaccia utente che renda la produzione del risultato desiderato semplice, efficace e divertente; questo, in genere, significa che l'utente deve fornire un contributo minimo per ottenere il risultato voluto ed anche che la macchina deve ridurre al minimo le richieste indesiderate.

In questo lavoro il problema di costruire un buon HMI si rifletterà nella progettazione della scheda di controllo: come si può intuire dall'esempio generico fornito in figura 2, il sistema comanderà un'interfaccia video per esaminare le

richieste effettuate e mostrare il loro risultato, un'interfaccia audio per scopi assistivi o semplicemente di intrattenimento, un'interfaccia tattile per impartire i comandi in modo intuitivo.



Figura 2: Un esempio di HMI privo dello chassis, con Android OS

Come è facile intuire dalle immagini 1 e 2, l'occupazione di area di questi sistemi è un fattore chiave, poiché spesso devono essere integrati sul dorso di un monitor, o in spazi piccoli e spesso con geometrie particolari; non è quindi possibile realizzarli con un normale personal computer, è necessario costruire un *sistema embedded* (dei quali parleremo in dettaglio subito al Capitolo 1).

La parte software dell'HMI, a differenza di quella hardware, verrà gestita dal committente del progetto: in figura 2 si può notare che il sistema operativo in uso sull'HMI è Android, il quale gestirà anche questo dispositivo.

Nel capitolo 2 è riportata l'illustrazione delle specifiche di progetto, lo schema a blocchi del sistema, la scelta della piattaforma su cui basarlo e la spiegazione delle varie parti di cui è composto; il capitolo 3 prende in esame lo stadio di alimentazione del sistema, illustrandone il funzionamento; nel capitolo 4 sono riportati gli schematici che costituiscono il progetto, mentre nel 5° capitolo mi sono soffermato ad illustrare tecniche di buon layout della board, evidenziando le principali problematiche di emissioni elettromagnetiche che un progettista può incontrare.

Capitolo 1

Panoramica sui sistemi embedded

Un computer è una macchina progettata per elaborare, archiviare e recuperare i dati, memorizzati sottoforma di numeri; è molto facile dimenticarsi di questo concetto, quando siamo abituati ad utilizzare tali macchine nell'uso quotidiano. Il computer manipola i dati per eseguire operazioni sui numeri; ad esempio, la visualizzazione di un'immagine su uno schermo si ottiene muovendo un'array nella memoria video, dove ogni numero rappresenta l'informazione associata al colore di un pixel, oppure, per riprodurre un file audio MP3, il computer legge un array di numeri dal disco e in memoria, manipola i numeri per convertire i dati audio dal formato compresso a quello di tipo raw ed invia il nuovo set di numeri (i dati audio grezzi) al chip audio.

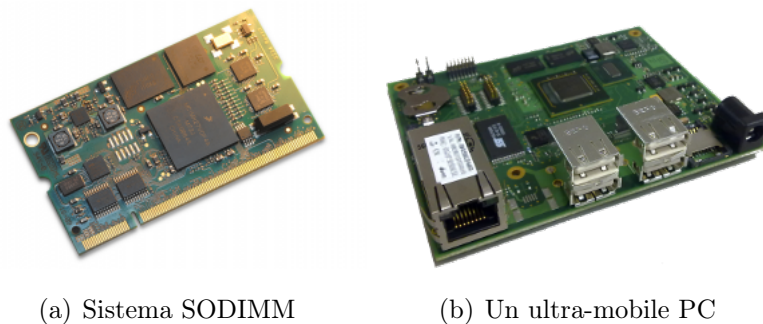
L'elettronica di un computer non è altro che un sistema (spesso di elevata complessità) progettato per contenere, spostare e cambiare i numeri.[1]

I sistemi *embedded* sono *sistemi di calcolo*, nel significato più generale del termine. Questa definizione, infatti, include tutti i computer del mondo, tranne quelli progettati per essere di utilità generica (“general purpose”).

Contrariamente ai computer generici riprogrammabili, un sistema embedded ha dei compiti noti già durante lo sviluppo, che eseguirà dunque grazie ad una combinazione hardware/software specificamente studiata per la tale applicazione. Dato il compito specifico che devono eseguire, l'hardware può essere ottimizzato al massimo al fine di ridurre il consumo di area e di potenza, i tempi di elaborazione ed il costo di fabbricazione. Inoltre l'esecuzione del software è spesso in tempo reale (real-time) per permettere un controllo deterministico dei tempi di esecuzione.

1.1 Esempi di sistemi embedded

Gli esempi di sistemi embedded spaziano dai lettori portatili di musica ai controlli in tempo reale di sistemi fisici quali lo Space Shuttle.



(a) Sistema SODIMM

(b) Un ultra-mobile PC

Figura 1.1: Esempi di sistemi embedded: sono tutti diversi fra loro

Portiamo qualche esempio a sostegno, per meglio comprendere la capillarità di tali sistemi nell'uso quotidiano:

- elettronica per l'aeronautica:
 - sistemi di guida inerziale;
 - hardware/software di controllo per il volo;
 - sistemi integrati nei velivoli e nei missili;
- elettronica per le telecomunicazioni e l'informatica:
 - centralini telefonici;
 - routers;
 - timeservers;
 - firewalls;
- elettronica per la domotica:
 - termostati;
 - condizionatori;

- sistemi per il monitoraggio della sicurezza;
- sistemi per l'analisi dei dati forniti da reti di sensori;
- sportelli Bancomat e apparecchi POS;
- distributori di bevande;
- apparecchiature biomedicali:
 - ecografi;
 - scanner per risonanza magnetica;
- strumenti di misura:
 - oscilloscopi digitali;
 - analizzatori di stati logici;
 - analizzatori di spettro;
- I PLC (Programmable Logic Controller) utilizzati per l'automazione industriale;
- elettronica di consumo:
 - console per videogiochi fisse e portatili;
 - telefoni cellulari;
 - stampanti e fotocopiatrici;
 - elettrodomestici:
 - * forni a microonde;
 - * lavatrici;
 - * apparecchi televisivi;
 - * lettori o masterizzatori di DVD;
 - Sistemi di stoccaggio di dati:
 - * hard disks;
 - * compact discs;
 - * flash drives;

- audio/video processing:
 - strumenti musicali digitali;
 - mixer digitali;
 - processori audio/video;
 - decoder per la gestione di flussi video digitali;
- elettronica industriale:
 - centraline di controllo delle autovetture e degli autocarri:
 - * motore;
 - * illuminazione adattiva del manto stradale;
 - * assistenza alla guida (ABS, ad esempio).

La diffusione di questi sistemi è capillare nella società e tutti i dispositivi elettronici non general purpose possono essere definiti sistemi embedded: sotto questa ottica, comprendiamo così la loro notevole importanza, ed il posto che occupano nella nostra vita quotidiana: il fatto che tali sistemi siano incorporati, dunque non visibili agli occhi dell'utilizzatore, induce il senso comune a pensare ad una prevalenza dei computer general purpose, quando, in realtà, la situazione è *diametralmente opposta*.

La maggior parte dei sistemi embedded è progettata per eseguire ripetutamente un'azione a costo contenuto; è inoltre loro richiesto di soddisfare dei vincoli di prestazione minime, come ad esempio la necessità di operare in tempo reale; può inoltre presentarsi l'eventualità che un sistema debba essere in grado di eseguire molto velocemente alcune predeterminate funzioni, ma possa tollerare velocità inferiori per altre attività: questi sistemi rispettano i vincoli di prestazione con una combinazione di hardware e software appositamente progettati.

Risulta difficile e probabilmente errato caratterizzare in modo generico la velocità o i costi di un sistema embedded, anche se, soprattutto per sistemi che devono processare una grande quantità di dati, il progetto stesso assorbe la maggior parte dei costi.

Per la maggior parte dei sistemi embedded le prestazioni richieste possono essere soddisfatte con una combinazione di hardware dedicato e una quantità

limitata di software ottimizzato. Spesso, gran parte dell'hardware di un sistema embedded deve sottostare a requisiti di prestazioni molto meno severi di quelli che, invece, deve rispettare l'hardware primario del sistema stesso. A titolo di esempio, basti pensare ad un decoder per una televisione satellitare: nonostante un sistema come questo debba processare decine di megabit di dati al secondo, la maggior parte del lavoro è svolta da hardware completamente dedicato che separa, regola e decodifica il flusso digitale multicanale in un'uscita video; alla CPU¹ embedded spetta determinare i percorsi dei dati nel sistema, o gestire gli interrupt, generare e disegnare la grafica, e così via. Questo permette all'architettura di un sistema integrato di essere semplificata rispetto a quella di un computer generico che deve eseguire le stesse operazioni, usando ad esempio una CPU più economica, ma che, tutto sommato, si comporta discretamente anche per queste funzioni secondarie; ciononostante, alcuni sistemi embedded potrebbero richiedere in ogni caso CPU potenti, hardware dedicato ed una grande quantità di memoria per eseguire una certa attività, magari molto onerosa dal punto di vista computazionale.

Nel caso di sistemi che devono essere commercializzati in massa, come un lettore di musica portatile, *ridurre i costi* diventa uno dei fattori chiave. Sistemi di questo genere, infatti, spesso sono dotati di alcuni chip, una CPU altamente integrata, un chip dedicato a tutte le altre funzioni ed un singolo banco di memoria: in questo caso, ogni componente è selezionato e progettato per ridurre il più possibile i costi.

Il software scritto per molti sistemi embedded, in particolare quelli senza hard disk, prende il nome di *firmware*; a titolo di esempio, è possibile trovare un firmware nei chip delle memorie ROM o in quelli delle memorie di tipo Flash. I sistemi embedded spesso richiedono di essere attivi *continuamente* per periodi di tempo molto lunghi (per mesi o anni, ad esempio) *senza incorrere in errori o malfunzionamenti*, pertanto il software ed il firmware sono progettati e testati con molta più attenzione rispetto al software dei personal computer. Molti sistemi embedded, infatti, evitano di incorporare componentistiche che possiedono parti meccaniche in movimento (come, ad esempio, gli hard disk drives), poiché risultano essere meno affidabili rispetto a componenti allo stato solido, come le

¹Central Processing Unit, l'unità centrale di elaborazione: è uno dei due componenti principali della macchina a programma memorizzato di von Neumann, il modello su cui è basata la maggior parte dei moderni computer.

memorie Flash, potendo così introdurre potenziali problematiche di robustezza del dispositivo (ed accorciando quindi l'MTBF).

In aggiunta, i dispositivi embedded possono essere *fisicamente inaccessibili* (come per le trivelle dei pozzi di petrolio, oppure i componenti lanciati nello spazio), pertanto i sistemi che li contengono devono essere in grado di resettarsi autonomamente in caso di perdita o corruzione dei dati, potendo quindi garantire di riprendere la loro funzionalità nel minor tempo possibile. Questa funzionalità è molto spesso ottenuta con l'inserimento di un componente elettronico chiamato timer watchdog, il quale ha il compito di ripristinare il processore se il programma che sta girando sullo stesso non azzerava il suo valore interno con una certa frequenza prefissata in fase di progettazione.

1.2 La piattaforma

La *piattaforma* sulla quale può venir sviluppato un sistema embedded, ovvero la base software e hardware su cui sono eseguite e sviluppate le applicazioni, varia drasticamente a seconda della sua complessità, dei consumi (elettrici), del costo e dell'ambito di utilizzo. Si passa dai PLC e microcontrollori più semplici ad architetture più complesse basate su circuiti integrati sofisticati (SoC²); come vedremo più avanti, quest'ultimo sarà il tipo di processore che verrà utilizzato per la realizzazione di questo progetto.

Alcune piattaforme di sviluppo (reference boards o reference designs) molto usate sono basate su architetture ARM, MIPS, Coldfire/68k, H8, SH, V850, FR-V, M32R, eccetera. Talvolta sono impiegate le più comuni architetture IBM compatibili, eventualmente adattate, con CPU X86 o PowerPC. Altre architetture più semplificate sono basate su microcontrollore PICmicro, Intel 8051, Atmel AVR. L'evoluzione, che porta ad una maggiore ingegnerizzazione dei componenti, si sta spostando, specie per sistemi embedded ad alto volume di produzione, verso i SoC: essi racchiudono, in un singolo circuito integrato di tipo ASIC³, il microcontroller/CPU e/o il DSP, memoria, oscillatori e clock, regolatore di tensione, eventuali interfacce AD/DA, nonchè verso l'esterno (USB, Ethernet,...).

Data la complessità crescente di tali piattaforme, è consueto che il produttore dell'hardware fornisca un BSP⁴ per semplificare il supporto e l'integrazione tra il software sviluppato ad hoc, l'ambiente operativo sottostante e l'hardware. Un altro comune metodo di progetto prevede l'utilizzo di FPGA⁵, con la programma-

²“System on a chip”: Il termine “System on a Chip” viene utilizzato per indicare quei particolari circuiti integrati che in un solo chip contengono un intero sistema, o meglio, oltre al processore centrale, integrano anche un chipset ed eventualmente altri controller, come quello per la memoria RAM, la circuiteria input/output o il sottosistema video.

³Acronimo per “Application Specific Integrated Circuit”, è un circuito integrato creato appositamente per risolvere un'applicazione di calcolo ben precisa. La specificità della progettazione, focalizzata sulla risoluzione di un unico problema, consente di raggiungere delle prestazioni difficilmente ottenibili con l'uso di soluzioni più generiche.

⁴Nei sistemi integrati, il “Board Support Package” è il codice di supporto di una specifica implementazione per una data scheda che sia conforme al dato sistema operativo; è comunemente costituito da un bootloader che contiene il minimo supporto periferiche per caricare il sistema operativo e dai device driver per tutte le periferiche della scheda.

⁵Un “Field Programmable Gate Array” è un circuito integrato digitale la cui funzionalità è programmabile via software.

zione di tutta la logica interna, inclusa la CPU; la maggior parte di FPGA sono progettate proprio per questo scopo. Tipicamente si fa uso di FPGA affiancandoli ad altri circuiti integrati per l'interfacciamento. Questa situazione è in contrasto con quella del mercato dei desktop computer, che al momento è composto solo da poche architetture concorrenti, principalmente Intel/AMD x86 e PowerPC di Apple/Motorola/IBM, quest'ultimo usato nei computer Apple Macintosh fino al 2005. Può risultare interessante menzionare anche lo standard PC/104 che, nonostante riguardi solamente il form factor (la taglia della scheda madre e simili) e il bus di comunicazione, è presente in ambito industriale e tipicamente impiega elettronica comune nei sistemi desktop (CPU X86) con riadattamenti per questi usi specifici. Usualmente, quindi, impiegano anche i medesimi sistemi operativi (principalmente DOS, Linux od un sistema operativo real-time, come ad esempio QNX, o Inferno).

1.3 Le interfacce utente

Le interfacce utente per i sistemi embedded variano anche di molto tra sistema e sistema e quindi meritano qualche commento aggiuntivo.

I progettisti di interfacce come PARC, Apple Computer, Boeing ed HP tendono a minimizzare il numero di diverse interazioni dell'utente. Ad esempio, i loro sistemi utilizzano due soli pulsanti (il minimo assoluto) per controllare un menu di sistema (un bottone verrebbe utilizzato per selezionare la successiva voce di menu, l'altro per attivare quella selezionata). Un *touch screen* oppure dei bottoni ai bordi dello schermo possono anche essere utilizzati per minimizzare le interazioni con l'utente.

Un altro truccetto molto semplice è quello di minimizzare e semplificare il tipo di output: ad esempio, a volte si usa associare un led allo stato di ogni interfaccia, oppure ad ogni situazione di errore; un'altra economica variazione è quella di creare due file di led, accoppiate ad una matrice di errori che possono verificarsi (l'utente può allora incollare delle etichette che spieghino più chiaramente l'errore).

I progettisti usano molto i *colori* e le sensazioni ad essi collegate: generalmen-

te, il rosso indica che vi è un problema, il giallo indica che vi è una condizione anormale, il verde indica che lo status è regolare; questa combinazione ricorda da vicino quella di un semaforo: molte persone sono in grado di comprenderla, anche istintivamente.

Molti progettisti fanno sì che l'interfaccia utente visualizzata cambi immediatamente dopo l'interazione con l'utente: se la macchina sta per eseguire qualche operazione, normalmente si avvia nel giro di pochi secondi, oppure fornisce un rapporto sul procedere dell'operazione.

Se è necessario uno schermo, molti progettisti si avvalgono di testo semplice, principalmente perché è un'interfaccia poco costosa, veloce da realizzare; un'interfaccia grafica, contrariamente, è gradevole alla vista e permette di fare qualunque cosa che l'interfaccia testuale propone, ma tipicamente aggiunge del ritardo a causa del flusso di progettazione necessario per realizzarla (ideazione, approvazione, traduzione) e, solitamente, necessita di uno o due programmatori aggiuntivi specializzati; inoltre, un'interfaccia troppo affollata potrebbe confondere gli utenti, perché può utilizzare simboli dall'aspetto poco familiare, oppure richiedere schemi di pensiero per il suo utilizzo meno intuitivi rispetto all'interfaccia testuale.

Se in un progetto è necessario riferirsi ad alcune parti fisiche della struttura (come in una fotocopiatrice), queste sono spesso etichettate con numeri ben visibili sullo chassis del prodotto. Una interfaccia di rete è semplicemente uno schermo remoto e si comporta grosso modo come una qualsiasi altra interfaccia utente. Una delle interfacce generiche basate su schermo più famose è quella composta da due bottoni ed una linea di testo nella lingua madre dell'utente. È utilizzata ad esempio nei cercapersone, nelle stampanti di fascia media, negli switch di rete ed in genere in tutte le situazioni che richiedono una elevata interazione con l'utente. Quando c'è del testo, il progettista sceglie una o più lingue: quella predefinita è normalmente una delle più conosciute dal gruppo di utenti a cui il prodotto è destinato. La maggior parte dei progettisti tende ad utilizzare il set di caratteri nativo della lingua, nonostante ciò possa rivelarsi difficoltoso. Le persone che utilizzano set di caratteri particolari preferiscono, infatti, leggere testi scritti in questa maniera. Il testo è normalmente tradotto da personale specializzato, anche se nello staff del progetto ci sono persone madrelingua. Un altro truccetto comune consiste nel rendere il più chiaro possibile sul display i metodi

di funzionamento del macchinario. L'interfaccia utente, solitamente, è sottoposta ad approvazione: spesso si tratta di un cliente, di un distributore oppure di qualcuno direttamente responsabile per la vendita del sistema; i committenti, infatti, tendono a prendere decisioni poco rapidamente, oppure a non prenderle affatto. Ciò causa costosi ritardi che potrebbero essere evitati come sopra riportato.

1.4 Sistema operativo

La presenza o meno di un completo sistema operativo su di un sistema embedded varia drasticamente a seconda della sua complessità architettonica e campo di utilizzo. Su microcontrollori semplici, tipicamente opererà ciclicamente un singolo programma di pochi byte senza alcuna sovrastruttura; su ambienti complessi possono trovare applicazione i medesimi sistemi operativi impiegati comunemente per scopi generali (Linux, Windows CE, ecc.) eventualmente personalizzati (per operare in un ambiente con risorse minimali, in gergo “a basso footprint”), oppure più specializzati per gestire eventi in sistema operativo real-time (come Vxworks o QNX) o estremamente specializzati e non disponibili sul mercato (come i sistemi operativi dei cellulari GSM di prima generazione, per esempio, sviluppati tipicamente dal produttore degli apparati stessi).

Capitolo 2

Progettazione del sistema

Il sistema che va controllato dalla scheda oggetto di questa tesi è rappresentato da una *macchina automatica dispensatrice di fluidi*; la macchina è in grado di conoscere ed erogare il prodotto richiesto dall'utente (o una serie di miscele dei prodotti richiesti) una volta che esso interagisce con l'interfaccia grafica specificatamente progettata per gestire le varie combinazioni di fluidi.

Scopo di questa tesi quindi è quello di provvedere all'*ammodernamento del sistema di controllo preesistente*, oramai obsoleto, della macchina, costruendo un sistema più moderno, privo di interruttori meccanici e fortemente orientato alla multimedialità.

Interagendo con lo schermo tattile, l'utente impartisce un ordine: la scheda si occupa di convertire tale comando in un segnale elettrico, codificato secondo uno standard proprietario, e di trasmetterlo tramite una interfaccia seriale all'attuatore, che si occuperà di effettuare concretamente l'operazione richiesta.

Il sistema dovrà essere sempre disponibile all'utente, quindi andranno considerati tutti gli accorgimenti necessari per evitarne il blocco, qualsiasi possa esserne il motivo.

In questo capitolo vengono descritte le fasi di progettazione del sistema, partendo dall'esposizione dell'applicazione, delle specifiche di progetto, le quali rappresentano il punto di inizio della discussione, poiché decretano gli ambiti in cui il progettista è vincolato nello sviluppo; verrà quindi ricavato uno schema a blocchi contenente tutte le parti che sono state dimensionate.

I punti successivi del capitolo prendono in esame la scelta della piattaforma e

dell'architettura: vengono presentate due Evaluation Boards, una di *Freescale*[®] (i.MX53 EVB) ed una di *Texas Instruments*[®] (AM389x EVM) e le motivazioni che hanno portato alla scelta della seconda.

2.1 Definizione delle specifiche di progetto

Dall'idea del sistema esposta nel paragrafo precedente è necessario passare ad una prima fase di definizione delle specifiche; le specifiche di progetto vengono usualmente concordate fra l'acquirente ed il costruttore del prodotto finito: il loro scopo è quello di dare forma ad un'idea al fine di renderla attuabile; spesso sono frutto di scelte di tipo prestazionale, economico, meccanico (ingombri, materiali impiegati, ...).

Vengono ora esposte solamente le specifiche tecniche inerenti al progetto; rimando il lettore a visionare il documento completo[3] per eventuali approfondimenti.

1. Specifiche tecniche di progetto:

- Processore: ARM[®] Cortex[™]-A8 RISC (operante alla frequenza di 1.5 GHz): il produttore ARM[®] ha raggiunto un notevole rapporto qualità/prezzo/prestazioni/per le applicazioni a basso consumo di potenza; dato il carico computazionale richiesto dal Flash Player di Adobe, è quindi una scelta preferenziale impiegare questa tipologia di processore per la realizzazione di un sistema embedded orientato alla multimedialità;
- Memoria di sistema: 1 Gbyte RAM DDR3: tale quantitativo di memoria di sistema risulta sufficiente per gestire il sistema operativo Android e l'applicazione Adobe Air; è stato considerato del margine, per evitare che, con aggiornamenti futuri, la RAM venga esaurita;
- Presenza di una porta seriale di tipo RS232 esportata su di un connettore non standard: la porta seriale ha lo scopo comandare l'attuatore della macchina dispensatrice di fluidi;
- Presenza di due porte USB 2.0 in configurazione "host": di queste porte, una è preposta alla connessione della schedina di controllo del touchscreen del monitor LCD, l'altra è dedicata all'aggiornamento del sistema operativo e/o dell'applicazione principale;

- Presenza di un'uscita audio di tipo stereo (Line out, non amplificata) esportata su di un connettore non standard: l'uscita di tipo audio ha il compito di guidare l'utente nell'utilizzo della macchina e di inviargli informazioni sullo stato di lavorazione;
- Presenza di uno slot interno che ospiti dei dispositivi di memoria di massa di tipo microSD: lo slot di tipo SD ospiterà una scheda di memoria di massa di tipo SecureDigital, la quale conterrà tutto il sistema operativo e l'applicazione principale (è il cosiddetto *boot medium*);
- Assorbimento medio di potenza della scheda: 15 W, comprensivo del consumo della retroilluminazione del display;
- Alimentazione DC: la scheda deve essere in grado di fornire l'alimentazione a tutto il sistema;
- MTBF¹ della scheda: 80000 ore.

2. Specifiche del display LCD:

- Area del display tipica: 220 (H) x 129 (V) mm;
- Risoluzione del display: 1024 (H) x 600 (V) Pixels;
- Luminanza tipica: $200 \frac{cd}{m^2}$;
- Consumo di potenza del display: inferiore ai 5 W.

3. Specifiche funzionali del pannello tattile di tipo resistivo:

- Tecnologia di costruzione del pannello: doppio strato resistivo di ITO² film;
- Materiale di costruzione del display: film di tipo ITO su vetro irrobustito con ITO;
- Area tipica attiva al tocco: 220x129 mm;
- Spessore tipico del pannello: 1,4 mm;
- Durata: > 1'000'000 di tocchi su di un singolo punto.

¹MTBF sta per "Mean Time Before Failure": il tempo medio fra i guasti è un parametro di affidabilità applicabile a dispositivi meccanici, elettrici ed elettronici e ad applicazioni software. Il MTBF è il valore atteso del tempo tra un guasto ed il successivo.

²L'ossido di indio-stagno (dall'inglese "Indium tin oxide") è il più impiegato ossido trasparente conduttivo per la fabbricazione di pellicole conduttive trasparenti (PCT).

4. Specifiche software:

- Sistema operativo: Linux 2.6.x, distribuzione specifica (Android);
- Avvio del sistema da una memoria di massa di tipo SecureDigital (SD o SDHC da 2 Gb o superiore) “in modalità a sola lettura”: non è permesso a nessun log ed a nessun dato creato dall’applicazione di rimanere salvato sul dispositivo in modo permanente;
- Supporto del display LCD specificato precedentemente tramite un driver video accelerato per il chipset di sistema;
- Possibilità di ruotare il display in modalità “portrait” quando il display opera in modalità grafica;
- Supporto per le applicazioni di tipo Adobe Air, versione 1.0;
- Avvio automatico dell’applicazione Flash a schermo intero; se l’applicazione principale si blocca, essa si ripristinerà automaticamente, non importa quale sia l’errore che ha causato il blocco;
- Un timer hardware di tipo watchdog riavvierà il sistema (dopo averne impostato il timeout, massimo 60s) se l’applicazione Flash si blocca per una qualsiasi motivazione.

2.2 Schema a blocchi del sistema

Nella prosecuzione dell'iter di progetto, dopo aver fissato le specifiche, cioè i limiti che vincolano l'ambito in cui il progettista può spaziare, mi sono concentrato sulla stesura di un diagramma a blocchi che schematizzi le componenti del sistema finale.

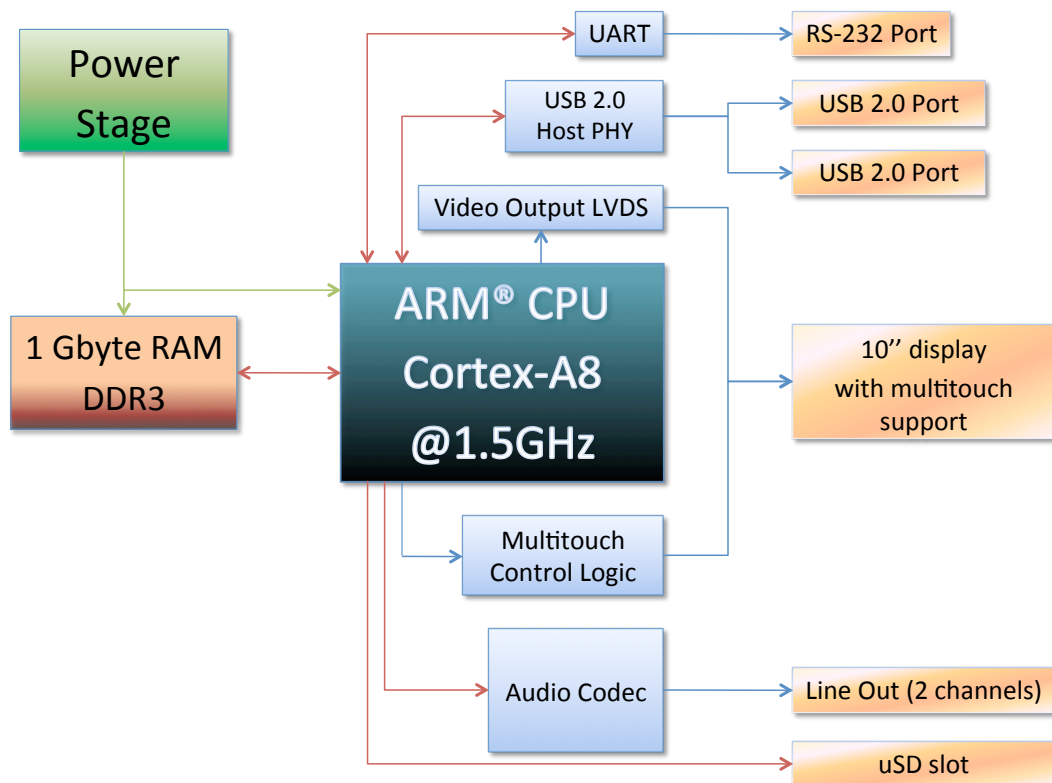


Figura 2.1: Schema a blocchi del sistema

Come si può vedere in figura 2.1, il sistema è composto dal processore centrale, dalle memorie, da tutte le periferiche richieste dal documento di specifica, dalla sezione di alimentazione e dai connettori direttamente disponibili all'utente.

2.3 Scelta dell'architettura e setup della piattaforma software

La scelta della piattaforma su cui basare il prodotto finale è influenzata da vari fattori: nella mia esperienza di tesi ho potuto rendermi conto che i due fattori predominanti sono il *livello di prestazione* offerta ed il *costo*; in realtà questi due fattori non sono scorrelati: il progettista deve sempre cercare di massimizzare le performance e, contemporaneamente, minimizzare i costi (per quanto è possibile). Per la scelta dell'architettura il discorso si limita alla constatazione che ARM[®] attualmente è in grado di offrire processori sempre più performanti, in grado di poter competere con leader del mercato quale è Intel[®], ad un prezzo *nettamente inferiore*; inoltre, i produttori di tali processori offrono BSP basati su sistemi operativi opensource (un esempio: i porting modificati di GNU/Linux per l'architettura ARM[®]), sulla base dei quali è possibile costruire rapidamente dei prodotti finiti senza incorrere necessariamente in costi aggiuntivi di licenze che possono minare la competitività commerciale del prodotto finale.

Avendo sviluppato il lavoro di tesi in Si14³, ho avuto il supporto dell'azienda per il reperimento di una serie di evaluation boards e di documentazione tecnica di riferimento utili per la valutazione del processore candidato all'inserimento nel progetto; i prodotti che dovevano essere studiati sono i seguenti, presentati in ordine cronologico:

- Pandaboard per il processore “OMAP4430” di Texas Instruments (una cpu dual-core ARM[®] Cortex[™]-A9 MPCore[™] con Symmetric Multiprocessing (SMP) operante alla frequenza di 1 GHz);
- EVK di Freescale Semiconductors per il multimedia application processor “i.MX536” (una cpu ARM[®] Cortex[™]-A8 operante alla frequenza di 1 GHz);
- EVM di Texas Instruments per il multimedia application processor “Sitara AM3892” (una cpu ARM[®] Cortex[™]-A8 operante alla frequenza di 1.5 GHz);

³<http://www.si14.com>

2.3.1 La piattaforma Pandaboard OMAP4430

Il processore offerto da Texas Instruments OMAP4430 è stato ideato appositamente per equipaggiare smartphones, tablet pc e dispositivi orientati alla multimedialità; è un processore veloce ma anche dai consumi contenuti, poiché questo è un parametro critico per questi dispositivi portatili; supporta in hardware flussi video Full HD 1080p ed è equipaggiato con una scheda grafica PowerVR™ provvista di accelerazione 3D, utile per costruire interfacce grafiche evolute; possiede inoltre un'unità di *content protection*, particolarmente utile in campo mobile, oltre che una nutrita serie di periferiche (USB 2.0 On-The-Go, interfacce seriali, MMC/SD, ...).

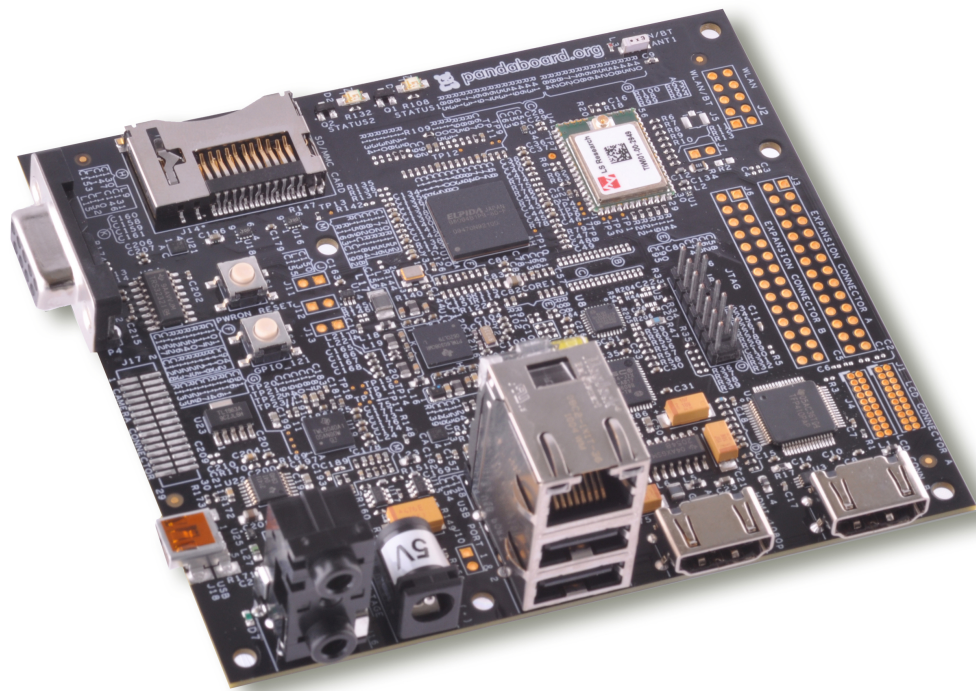


Figura 2.2: Pandaboard Evaluation Board

Non mi soffermo eccessivamente nel trattare questo prodotto perché, nella fase di studio di fattibilità, ci si è resi conto che, dato il target specifico di questo dispositivo (il mondo mobile), il processore supporta solamente memorie esterne

di tipo Low-Power DDR2 PoP. Le memorie Package-on-Package sono state create per ridurre le dimensioni fisiche del sottosistema di memoria su un computer single-board; l'idea di base è quella di impilare due dispositivi BGA⁴ uno sopra l'altro, come indicato in figura 2.28:

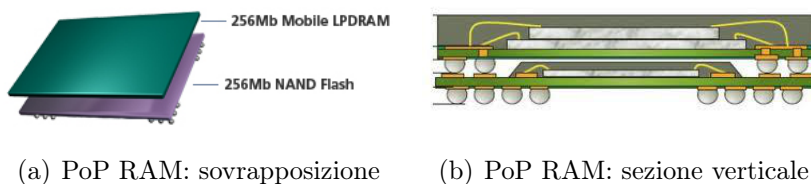


Figura 2.3: Le memorie RAM di tipo Package-on-Package

Questa tipologia di memorie di per sè sarebbe ottimale, però i produttori contrattano le vendite solamente per clienti in grado di richiedere *un numero ingente di pezzi*, molto superiore al target di progetto; per questa motivazione Panda-board, sebbene sia di per sè performante e molto interessante, è stata ritenuta non soddisfacente ai requisiti di progetto.

⁴BGA sta per *Ball Grid Array*, un metodo di contattazione fra scheda e circuito integrato.

2.3.2 La piattaforma Freescale[®] i.MX53

Nella sua gamma di prodotti Freescale[®] offre un processore Cortex-A8, di tipo SoC ad alto livello di integrazione, orientato alla multimedialità: supporta in hardware flussi video HD 1080p, è presente il supporto per Adobe[®] Flash[®] richiesto dalla specifica di progetto; il processore lavora ad una frequenza di core di 1 GHz; riporto per completezza le caratteristiche di interesse:

1. Caratteristiche hardware:

- CPU: ARM Cortex-A8 up to 1GHz;
- Advanced power management features;
- OpenGL[®] ES 2.0 and OpenVG 1.1 hardware accelerators;
- Multi-format HD1080p video decoder and HD720p video encoder hardware engine;
- Up to 2GB DDR2 and DDR3 SDRAM, 16/32-bit, 400 MHz;
- Managed NAND Flash Support with eMMC 4.4/SDIO;
- Raw NAND with up to 16-bit ECC;
- Dual display capable with multiple display options including TFT LCD, LVDS, analog TV-formats (composite, component, RGB) and standard VGA;
- Hardware accelerated image post-processing, display quality enhancement, and video and graphics combining;
- Two simultaneous camera inputs with hardware pre-processing;
- Dual USB 2.0 Controllers (HS OTG, HS Host) with integrated PHY;
- Two additional High-Speed USB 2.0 controllers;
- SATA controller and PHY up to 1.5Gbps;
- 10/100 Ethernet controller with IEEE1588 time-stamping;
- Wide array of serial interfaces including SDIO, SPI, I2C, UART;
- Security solution supporting High Assurance Boot, Cipher and random number generator accelerators, and Tamper Detection.

2. Caratteristiche software:

2 PROGETTAZIONE DEL SISTEMA

- BSPs available for all major smart mobile device OSes, including Android™, Chromium™, Windows® Embedded Compact 7, Linux®;
- Fully-optimized and hardware-accelerated Adobe Flash Player 10.1;
- Extensive portfolio of dedicated hardware-accelerated and NEON-optimized multimedia codecs.

Evaluation Board del processore i.MX536

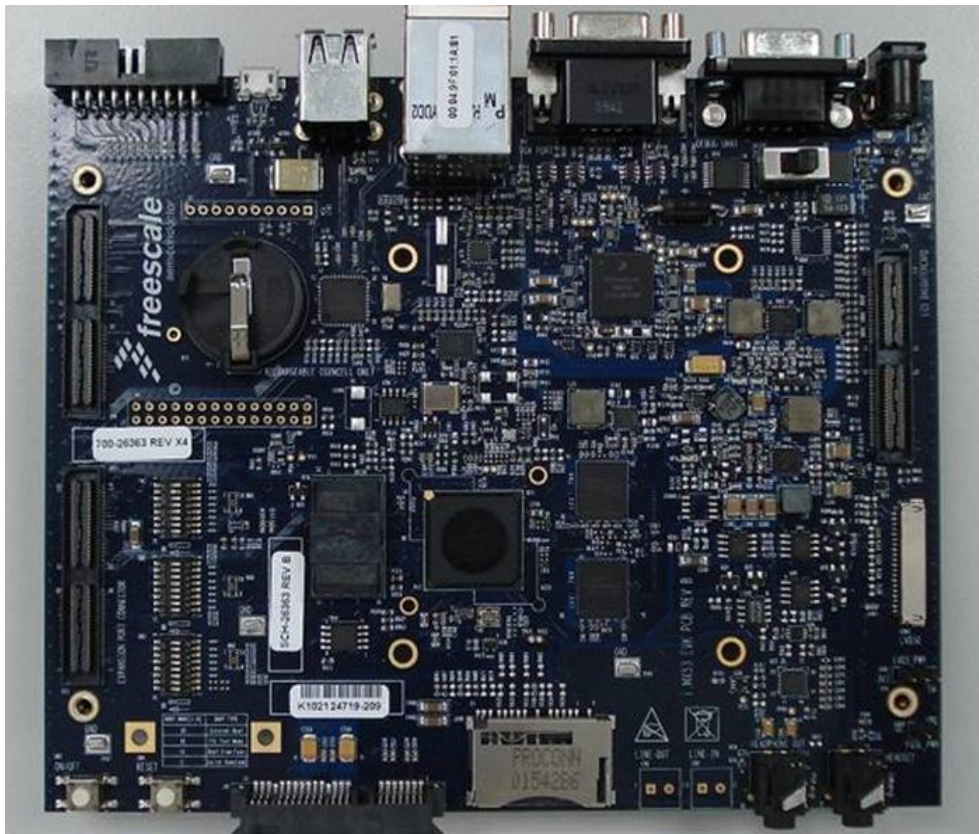


Figura 2.4: i.MX536 Application Processor Evaluation Kit

Setup del BSP Android® e della Main Application

La main application su cui il sistema si basa è stata realizzata tramite Adobe AIR: Adobe Integrated Runtime (AIR) è un ambiente di sviluppo multiplatforma per applicazioni internet che utilizzano Adobe Flash, Adobe Flex, HTML, o

AJAX, che possono essere utilizzate come applicazioni desktop; questo ambiente di runtime, senza alcun uso di un browser, consente di costruire applicazioni garantendo molte delle caratteristiche dei più tradizionali programmi installabili su desktop, grazie all'uso di codice Flash, ActionScript, HTML e JavaScript; ai fini di progetto, AIR risulta essere solo un “contenitore” per la vera applicazione, rappresentata da una pura Flash App.

Il mio obiettivo è stato quello di testare le performance dell'applicativo finale: questo ha comportato il predisporre tutta la struttura software necessaria al fine di raggiungere l'obiettivo; questo step, di fatto, non è propriamente una mansione di progettazione, ma è un chiaro esempio del fatto che una conoscenza prettamente settoriale spesso non è sufficiente per ottenere i propri obiettivi.

Il lavoro di preparazione dell'ambiente software ha seguito questo iter, descritto in 10 punti salienti:

1. Scaricamento dei sorgenti di bootloader U-boot, kernel Linux Freescale, sistema operativo imx-Android-r9.2 dalle repository Freescale;
2. cross-compilazione⁵ Bootloader U-boot;
3. cross-compilazione kernel Linux;
4. cross-compilazione sistema operativo imx-Android-r9.2;
5. creazione di una volume di tipo SecureDigital con un partizionamento ad-hoc per l'avvio del sistema;
6. test della correttezza e integrità della mappa delle partizioni;
7. configurazione del bootloader con le flags corrette, necessarie per attivare le opportune periferiche;
8. avvio del sistema;

⁵La cross-compilazione è la tecnica mediante la quale si compila un codice sorgente con un cross-compilatore, ottenendo così un file binario eseguibile su di un elaboratore *con architettura diversa* da quella della macchina su cui si è lanciato il cross-compilatore stesso.

9. predisposizione di ADB (ovvero *Android Debug Bridge*⁶) e, tramite questo tool, installazione con i privilegi di root⁷ di Adobe Flash Player;
10. installazione, tramite ADB, dell'applicativo finale.

Test ed aderenza della piattaforma alle specifiche di progetto

Dopo una opportuna configurazione dell'ambiente software, non sono riuscito ad impostare correttamente l'output video sull'interfaccia LVDS: dopo un attento studio dei parametri di configurazione del bootloader per l'abilitazione del driver, ho scoperto che questo problema è dovuto al fatto che la versione preliminare fornitami dell'ambiente (cioè la *imx-android-r9.2*) non aveva al suo interno ancora implementato correttamente il driver per LVDS; va precisato che gli altri test ho potuto effettuarli via terminale linux, tramite un accesso via porta seriale, o tramite un touchscreen che veniva offerto dal kit del prodotto.

Motivazioni che hanno condotto al cambio di piattaforma

La piattaforma i.MX53 è risultata piuttosto convincente dal punto di vista hardware, ma non completamente soddisfacente, poiché la capacità di calcolo non è stata ritenuta sufficiente in caso di un possibile sviluppo futuro dell'applicazione finale (un'aumento di complessità, e quindi di carico computazionale sul processore).

⁶Android Debug Bridge è uno strumento che viene fornito con il Software Developing Kit di Android che consente di controllare e di interfacciarsi con il dispositivo Android.

⁷Nei sistemi *nix, l'Amministratore di sistema (root o superuser) è l'utente che possiede i più elevati privilegi.

2.3.3 La piattaforma Texas Instruments® Sitara

Texas Instruments, con la sua piattaforma Sitara (processore AM3892), basata sul core ARM, offre una valida combinazione fra performance, integrazione per le telecomunicazioni, funzioni di networking, single-board computer SoC e automazione industriale; le MPU di questa fascia offrono prestazioni elevate e operano con una frequenza di core (che è un Cortex-A8 prodotto da ARM) fino a 1.5 GHz. Questo microprocessore consente di variare il consumo di potenza in funzione del carico computazionale, al fine di consentire una ottimizzazione, basata strettamente sulla specifica applicazione, dei costi e del consumo di potenza.

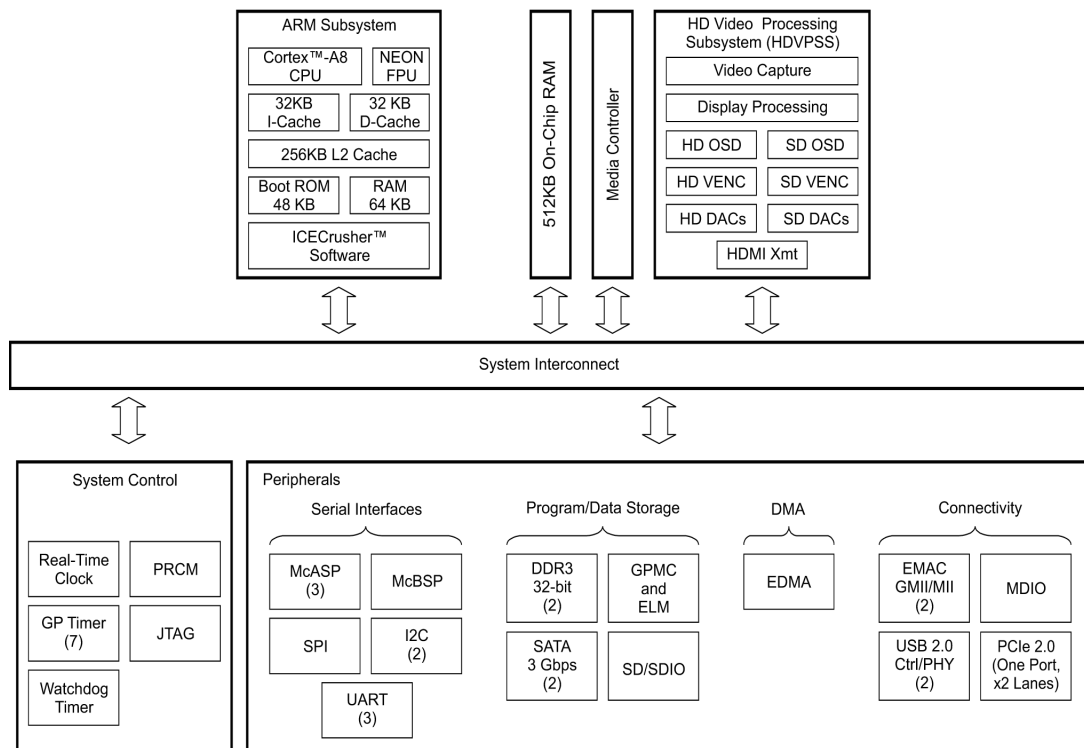


Figura 2.5: Schema a blocchi funzionale del processore AM3892

Come si può osservare dalla figura 2.5, la dotazione di periferiche integrate è notevole: spiccano il bus PCI Express, il bus USB 2.0, il bus SATA 2.0, il controller Ethernet di classe Gigabit, le interfacce verso la memoria esterna di tipo DDR2 e DDR3; il processore offre il supporto per un doppio display grazie ad un motore di visualizzazione on-chip, per l'output simultaneo di due diversi flussi di contenuti full-HD (alla risoluzione 1920 x 1280). Riporto per completezza le

caratteristiche di interesse:

- ARM® Cortex™-A8 RISC MPU
 - ARMv7 Architecture
 - Supports Integer and Floating Point (VFPv3-IEEE754 compliant)
- ARM® Cortex™-A8 Memory Architecture
 - 32K-Byte Instruction and Data Caches
 - 256K-Byte L2 Cache
 - 64K-Byte RAM, 48K-Byte Boot ROM
 - 512K-Bytes On-Chip Memory Controller (OCMC) RAM
- ARM Instructions/Data: Little Endian
- HD Video Processing Subsystem (HDVPSS)
- Two 165 MHz HD Video Display Channels
- Simultaneous SD and HD Analog Output
- Dual 32-bit DDR2/3 SDRAM Interfaces
 - Supports up to DDR2-800 and DDR3-1600
 - Up to Eight x8 Devices Total
 - 2 GB Total Address Space
 - Dynamic Memory Manager (DMM)
 - * Programmable Multi-Zone Memory Mapping and Interleaving
 - * Enables Efficient 2D Block Accesses
 - * Supports Tiled Objects in 0°, 90°, 180°, or 270 Orientation and Mirroring
 - * Optimizes Interlaced Accesses
- Dual USB 2.0 Ports With Integrated PHYs
 - USB 2.0 High-/Full-Speed Client

- USB 2.0 High-/Full-/Low-Speed Host
- General Purpose Memory Controller (GPMC)
 - 8-/16-bit Multiplexed Address/Data Bus
 - Up to 6 Chip Selects With up to 128M-Byte Address Space per Chip Select Pin
 - Glueless Interface to NOR Flash, NAND Flash (With BCH and Hamming Error Code Detection), SRAM and Pseudo-SRAM
 - Error Locator Module (ELM) Outside of GPMC to Provide Up to 16-Bit/512-Bytes Hardware ECC for NAND
 - Flexible Asynchronous Protocol Control for Interface to FPGA, CPLD, ASICs, etc.
- Enhanced Direct-Memory-Access (EDMA) Controller
 - Four Transfer Controllers
 - 64/8 Independent DMA/QDMA Channels
- Seven 32-bit General-Purpose Timers
- One System Watchdog Timer
- Three Configurable UART/IrDA/CIR Modules
 - UART0 With Modem Control Signals
 - Supports up to 3.6864 Mbps UART
 - SIR, MIR, FIR (4.0 MBAUD), and CIR
- One 40-MHz Serial Peripheral Interface (SPI) With Four Chip-Selects
- SD/SDIO serial interface (1-/4-Bit)
- Dual Inter-Integrated Circuit (I2C BUS®) Ports
- Three Multichannel Audio Serial Ports
- Multichannel Buffered Serial Port (McBSP)
 - Transmit/Receive Clocks up to 48 MHz

- Two Clock Zones and Two Serial Data Pins
- Supports TDM, I2S, and Similar Formats
- Real-Time Clock (RTC)
- Up to 16 General-Purpose I/O (GPIO) Pins
- On-Chip ARM® ROM Bootloader (RBL)
- Power, Reset, and Clock Management
 - SmartReflex™ Technology (Level 2)
 - Seven Independent Core Power Domains
 - Clock Enable/Disable Control For Subsystems and Peripherals
- IEEE-1149.1 (JTAG) and IEEE-1149.7 (cJTAG) Compatible
- 1031-Pin Pb-Free BGA Package (CYG Suffix), 0.65-mm Ball Pitch
- 40-nm CMOS Technology

Evaluation Board del processore AM3892

L'evaluation board fornitami da Texas Instruments non è stata ideata specificamente per il processore Sitara AM3892 di mio interesse, bensì per il processore Integra: questo non è un problema, poiché Integra non è altro che un Sitara AM3894 con affiancato un DSP operante alla frequenza di 1 GHz⁸; infine, il processore AM3894 è un core AM3892 con all'interno un engine grafico 3D PowerVR™ SGX530; la piattaforma, quindi, una volta che è stata correttamente configurata, rende possibile utilizzare solamente il core AM3892.

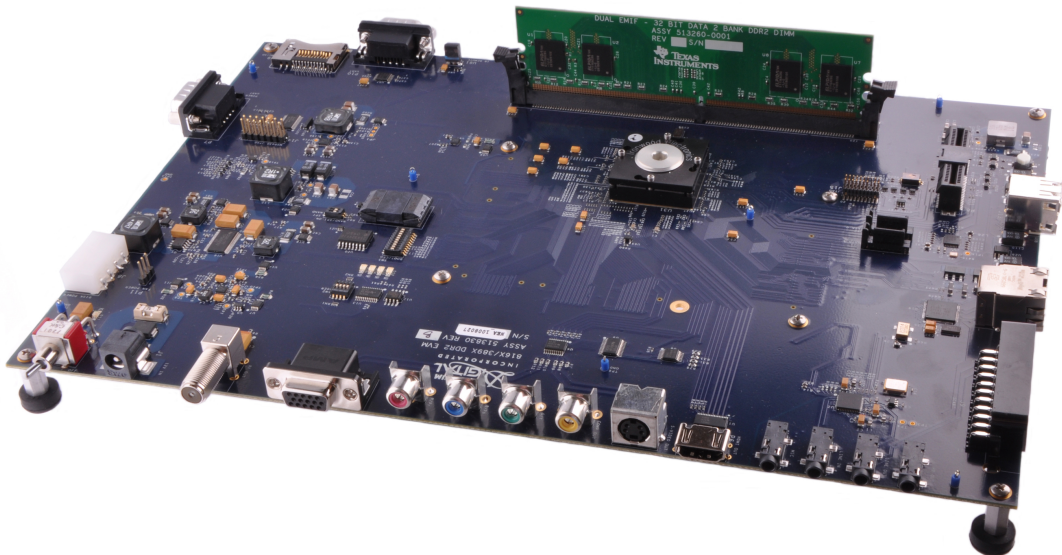


Figura 2.6: AM3892 Application Processor Evaluation Module

Setup del BSP Android® e della Main Application

Come ho precedentemente esposto a riguardo della board di Freescale, l'obiettivo è stato quello di testare le performance dell'applicativo finale sul prodotto Sitara; è stato necessario predisporre tutta la struttura software necessaria. Un'aspetto interessante che ho potuto constatare è che, nel mondo dei sistemi embedded, ogni produttore segue la sua filosofia, la sua strada, i propri standard: mi aspettavo di dover impostare una configurazione simile a quella costruita per i.MX53,

⁸il DSP opera alla stessa frequenza operativa del processore.

invece non è stato possibile. Il lavoro di preparazione dell'ambiente software ha comunque seguito un iter descrivibile sinteticamente in 10 punti salienti:

1. Scaricamento dei sorgenti di bootloader U-boot, kernel Linux Texas, sistema operativo Android Rowboat dalle repository Texas Instruments;
2. cross-compilazione⁹ Bootloader U-boot;
3. cross-compilazione del kernel Linux modificato Texas;
4. cross-compilazione sistema operativo Android Rowboat;
5. creazione di una volume di tipo SecureDigital con un partizionamento ad-hoc per l'avvio del sistema;
6. test della correttezza e integrità della mappa delle partizioni;
7. configurazione del bootloader con le flags corrette, necessarie per attivare le opportune periferiche;
8. avvio del sistema;
9. predisposizione di ADB (ovvero *Android Debug Bridge*¹⁰) e, tramite questo tool, installazione con i privilegi di root¹¹ di Adobe Flash Player;
10. installazione, tramite ADB, dell'applicativo finale.

Test ed aderenza del prodotto alle specifiche di progetto

Dopo una opportuna configurazione dell'ambiente software, non ho potuto utilizzare l'output video sull'interfaccia LVDS poiché essa *non è disponibile direttamente all'utente*: è invece esportata l'uscita parallela RGB su di un connettore sul lato posteriore della board: da qui si è pensato di effettuare una conversione parallelo-seriale (tramite lo sviluppo di una *piccola board di adattamento* esterno)

⁹La cross-compilazione è la tecnica mediante la quale si compila un codice sorgente con un cross-compilatore, ottenendo così un file binario eseguibile su di un elaboratore *con architettura diversa* da quella della macchina su cui si è lanciato il cross-compilatore stesso.

¹⁰Android Debug Bridge è uno strumento che viene fornito con il Software Developing Kit di Android che consente di controllare e di interfacciarsi con il dispositivo Android.

¹¹Nei sistemi *nix, l'Amministratore di sistema (root o superuser) è l'utente che possiede i più elevati privilegi.

per costruire l'interfaccia LVDS richiesta dalla specifica di progetto.

Gli altri test ho potuto effettuarli via terminale linux, tramite un accesso via porta seriale, sfruttando Minicom, oppure tramite l'interfaccia VGA che veniva offerta dalla board; sono rimasto stupito del fatto che, nella sua EVK, Texas ha reso disponibile molte diverse connessioni (HDMI, VGA, CVBS, Component, S-Video, SCART) tralasciandone alcune molto utilizzate in ambito di sistemi embedded (RGB, LVDS): questo fatto, comunque, non ha inficiato la scelta del prodotto, poiché è stato comunque possibile raggiungere lo scopo prefisso.

Motivazioni che hanno condotto alla scelta della piattaforma

La piattaforma Sitara è risultata convincente sotto ogni punto di vista; la EVM ha dato buoni risultati alla frequenza operativa di 1.0 GHz (che di per sè non sono stati comunque ritenuti sufficienti); il prodotto finale opererà a 1.5 GHz, riservando un sufficiente margine di potenza di calcolo in caso di modifiche future della main application.

2.4 Studio del consumo di potenza

2.4.1 I domini di tensione del sistema

Un *dominio* è una sezione di alimentazione del dispositivo che possiede una gestione del flusso di potenza indipendente e dedicata; un power domain può essere attivato e disattivato senza influire sulle altre parti del dispositivo; lo scopo dei power domains è quello di minimizzare il consumo di potenza del circuito.

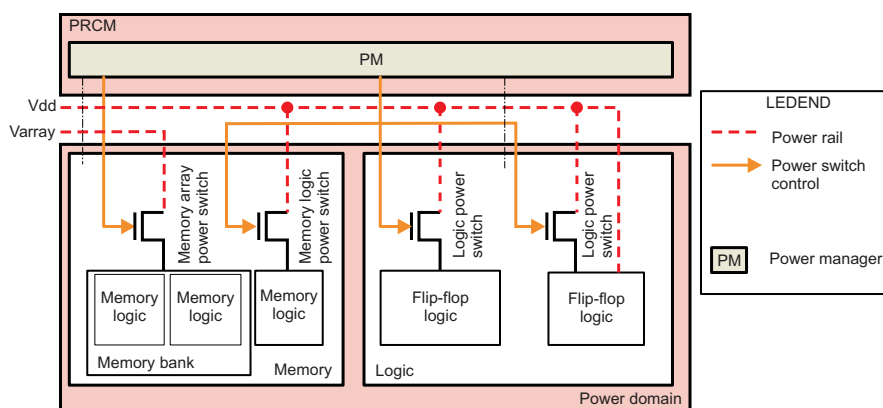


Figura 2.7: Struttura di un dominio di alimentazione

Adaptive Voltage Scaling (AVS)

Adaptive Voltage Scaling (AVS) è una tecnica di power-management basata su Smart Reflex che viene utilizzata per il controllo automatico delle tensioni di funzionamento del dispositivo, al fine di ridurre il consumo di potenza attiva. Con Smart Reflex, la tensione di alimentazione viene adeguato alle prestazioni offerte dal silicio, in modo statico (basato su punti di lavoro predefiniti nel processo di fabbricazione di un determinato dispositivo) o dinamico (in base alle performance indotte in tempo reale dalla temperatura del dispositivo). Un confronto di questi punti di lavoro predefiniti con la misura delle prestazioni in tempo reale del chip determina se sia necessario innalzare o abbassare la tensione di alimentazione.

I domini di tensione 1V-AVS e 1V-Costante possiedono sette sottodomini di potenza che alimentano sia la logica sia le SRAM interna ai rispettivi moduli;

tutti gli altri domini di tensione hanno la caratteristica di essere sempre attivi (“always on”). All’interno di questi due particolari domini sono presenti degli interruttori interni dell’alimentazione che sono in grado di interrompere completamente la fornitura di potenza ai sottodomini. All’avvio del sistema tutti i settori, ad eccezione di quello always-on, vengono abilitati quando ricevono un opportuno segnale; essendoci un sottodominio di tipo always-on in ciascun dominio di tensione, si presuppone che tutti gli alimentatori siano essere accesi per tutto il tempo, fino a quando il dispositivo è in funzione.

Domini 1V-AVS e 1V-Costante

- Domino “Active”: il dominio *active* ha al suo interno tutti i moduli che sono richiesti nello stato “attivo” del processore; i moduli qui presenti non sono richiesti in nessuno stato di standby; qui risiede la periferica HDVPSS.
- Domino “Default”: il dominio *default* contiene tutti quei moduli che potenzialmente possono essere richiesti anche in un qualsiasi stato di standby; la possibilità di avere dei moduli in questo dominio consente di poter abilitarli e disabilitarli anche se il processore è in uno stato di standby; qui risiedono le periferiche DDR, SATA, PCI Express, il Media Controller e le USB.
- Domino “Always-on”: il dominio *always-on* contiene tutti quei moduli che **sono** richiesti anche in un qualsiasi stato di standby; qui risiedono il core ARM, gli I/Os a basso consumo di potenza e tutti i moduli preposti alla generazione di Interrupts¹²(UART, RTC, GPIO, EMAC).

La tecnologia SmartReflex

Il dispositivo contiene i moduli SmartReflex, i quali sono necessari per ridurre al minimo il consumo di energia sui domini utilizzando una sorgente di alimentazione a tensione variabile. I moduli SmartReflex, basandosi sulla temperatura e sulle performance richieste, istruiscono il processore su come innalzare o abbassare la tensione di alimentazione per ciascun dominio al fine di avere un consumo energetico minimo. Il canale di comunicazione tra il processore centrale ed i regolatori esterni può essere realizzato utilizzando dei GPIO, il bus I²C o il bus

¹²Ovvero interruzioni più o meno critiche del ciclo di esecuzione del programma nel sistema embedded.

SPI. La tecnica principale impiegata dalla tecnologia SmartReflex nel dispositivo è la AVS (vedi paragrafo 2.4.1: essa è attiva di continuo e agisce in tempo reale, aiutando a minimizzare il consumo energetico in risposta alle modifiche delle condizioni di funzionamento.

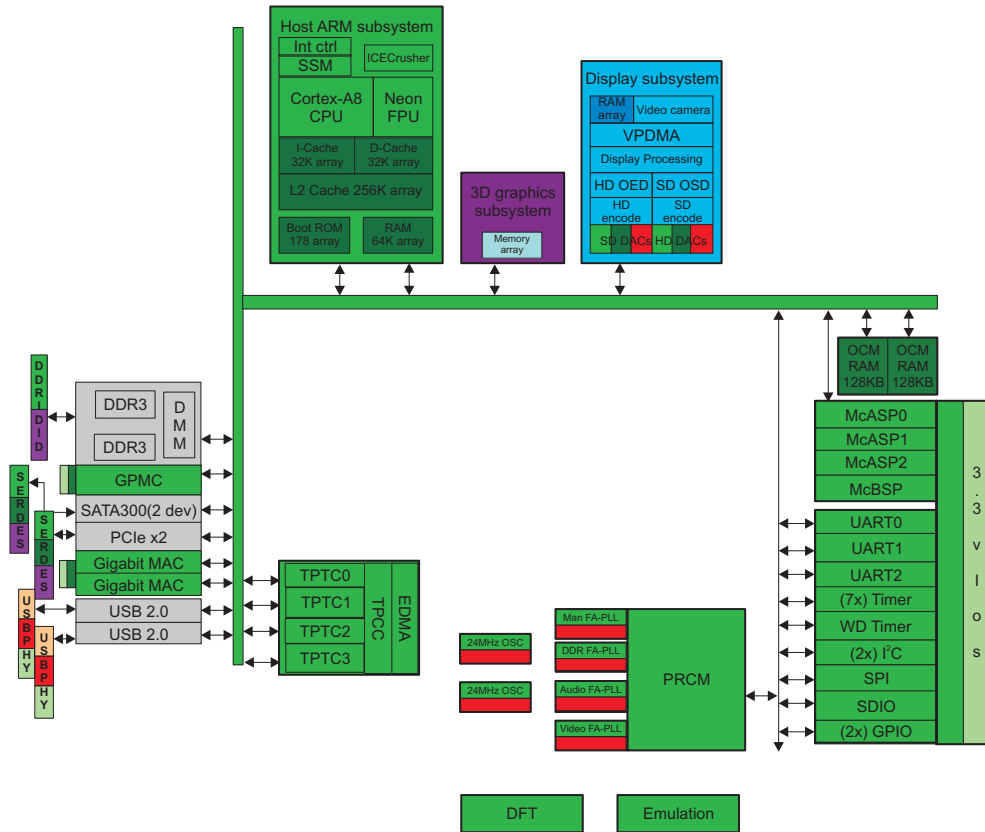
2.4.2 Gestione del risparmio energetico per le memorie

Il dispositivo fornisce una funzionalità che permette al software di mettere le memorie a livello di chip (OCMC RAM) in tre stati; queste tre diverse modalità per risparmiare energia quando le memorie non sono utilizzate sono qui descritte:

- *light sleep* (LS): in questa modalità, il risparmio energetico raggiunge circa il 60%, la latenza per il riavvio è bassa e i dati permangono nella memoria;
- *deep sleep* (DS): in questa modalità, il risparmio energetico raggiunge circa il 75%, la latenza per il riavvio è media e i dati permangono comunque nella memoria;
- *shutdown* (SD): in questa modalità, il risparmio energetico raggiunge circa il 95%, la latenza per il riavvio è alta e i dati vengono persi.

Al fine di ridurre le correnti di perdita nelle SRAM è possibile commutare dalla modalità attiva a quella di shutdown blocchi di SRAM; quando tali blocchi sono in modalità a basso consumo, la tensione che li alimenta viene automaticamente disconnessa e tutti i dati che ivi sono immagazzinati vengono perduti.

Tutte le memorie di tipo SRAM in un dominio di potenza “commutabile” (quindi tutti i domini, tranne gli *always-on*), entrano automaticamente in modalità di arresto ogni volta che il dominio a loro assegnato passa allo stato di OFF; viceversa, le SRAM ritornano allo stato attivo quando il ritorno il dominio corrispondente ritorna allo stato di ON.



	1V AVS	1V	1.8V	3.3V	1.5V	0.9V
Always-on						
Default						
Active						
SGX						

Figura 2.8: Domini delle tensioni di alimentazione

2.5 La sequenza di avvio del sistema

Data la complessità del sistema, affinché esso possa avviarsi correttamente è necessario rispettare la precisa sequenza di avviamento prevista dal costruttore. Gli alimentatori del sistema devono avviarsi in questo ordine:

1. 12V;
2. 5V;
3. 3.3V;
4. 1V-AVS;
5. 1V-Costante;
6. 1.8V;
7. 1.5V;
8. 0.9V.

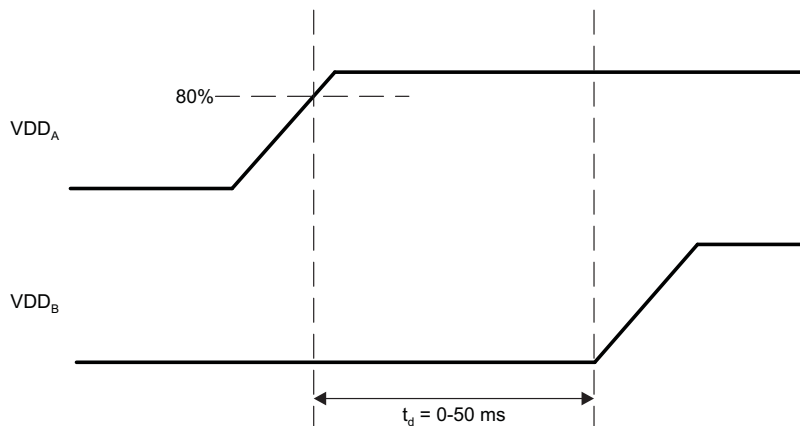


Figura 2.9: Requisiti per l'avvio sequenziale delle tensioni di alimentazione

Come illustrato nella figura 2.9, ogni alimentazione¹³ (rappresentata da VDD_B) deve avviarsi entro un tempo t_d che va da 0 ms a 50 ms dopo che l'alimentazione precedente nella sequenza (rappresentata da VDD_A) ha raggiunto l'80% del suo valore nominale[4].

¹³Il discorso proseguirà alla sezione 3.2 quando si dimensionerà lo stadio di alimentazione.

2.6 I clock di sistema

I clock di sistema sono gestiti dal modulo demonimato PRCM¹⁴, che si occupa del loro pilotaggio; I clock sono gestiti in base ai vincoli imposti dai moduli a loro associati: sulla base di queste caratteristiche essi sono divisi in due categorie: clock di interfaccia e clock funzionali.

I clock di interfaccia hanno le seguenti caratteristiche:

- garantiscono una corretta comunicazione tra qualsiasi modulo / sottosistema e l'interconnessione;
- nella maggior parte dei casi, temporizzano l'interfaccia di interconnessione al sistema ed i registri del modulo;
- tipicamente un modulo ha un clock di interfaccia, ma esistono moduli con più clock di interfaccia;
- la gestione dell'interfaccia di clock è compiuta a livello di dispositivo.

I clock funzionali hanno le seguenti caratteristiche:

- temporizzano la parte funzionale di un modulo o di un sottosistema;
- un modulo può avere uno o più clock funzionali: alcuni di essi sono necessari, mentre altri sono facoltativi;
- un modulo necessita dei clock funzionali obbligatori per essere attivo; i clock opzionali, invece, sono utilizzati per caratteristiche specifiche e possono essere arrestati senza bloccare l'attività globale modulo.

I clock di sistema sono generati da diversi clock esterni di riferimento che sono alimentati da PLL on-chip e divisori (sia all'interno che all'esterno del modulo PRCM); la figura 2.10 mostra una panoramica della struttura del clock manager:

¹⁴sta per "Power Reset Clock Manager", ovvero il gestore della sezione di potenza, di reset e di clocking del dispositivo.

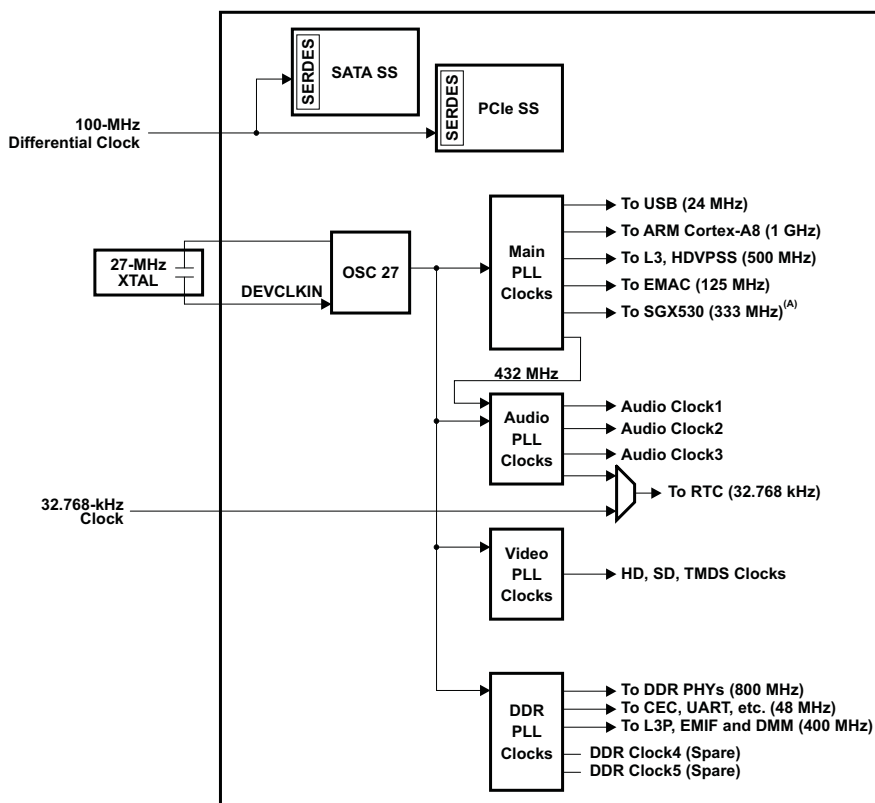


Figura 2.10: Il clock manager

I condensatori di carico C_1 e C_2 (mostrati in Figura 2.11), devono essere scelti in modo da soddisfare la seguente equazione:

$$C_L = \frac{C_1 \cdot C_2}{C_1 + C_2} \quad (2.1)$$

C_L nell'equazione 2.1 è il carico specificato dal produttore del cristallo; R_d (vedi figura 2.11) è un resistore opzionale smorzamento; è importante fra presente che tutti i componenti discreti utilizzati per realizzare il circuito oscillatore devono essere collocati *il più vicino possibile* ai pin ad esso associati (MXI, MXO, VSS).

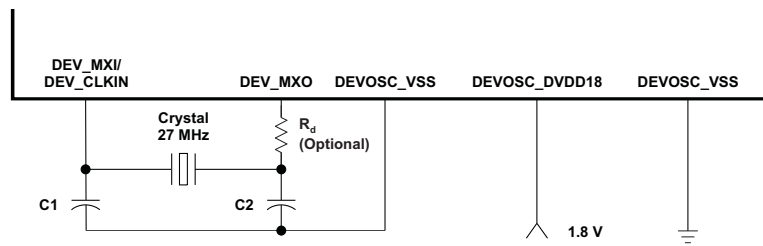


Figura 2.11: Il clock a 27 MHz

2.7 I bus e le periferiche di sistema

In questa sezione vengono presentati un pò più in dettaglio le componenti del sistema che sono state impiegate nello schematico.

2.7.1 I Timer

Il modulo dei timer contiene dei contatori capaci di contare verso l'alto, provvisti di auto-reset in caso di overflow¹⁵; tale modulo contiene della logica di comparazione per consentire di lanciare un evento di interruzione, una volta raggiunto un valore di conteggio prefissato.

Il contatore del timer può essere letto e scritto in tempo reale, anche mentre sta avvenendo un conteggio.

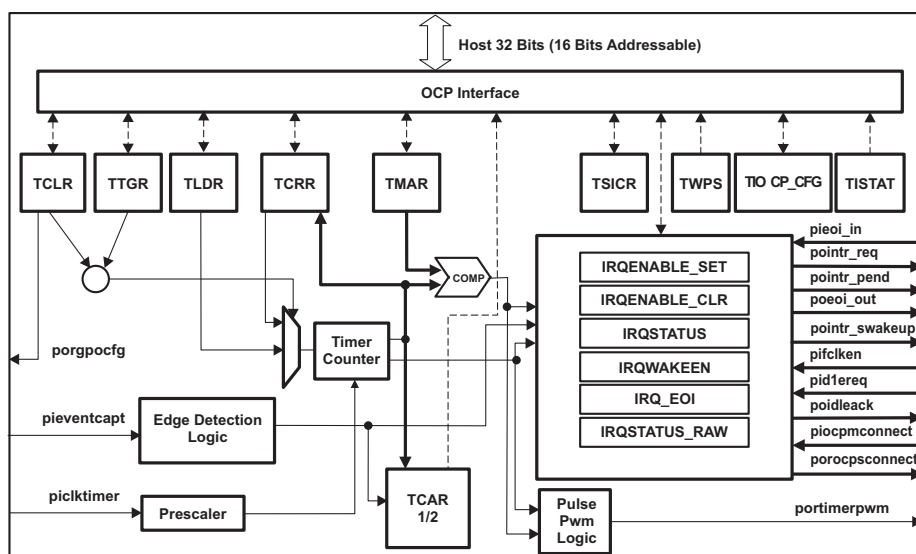


Figura 2.12: Schema a blocchi del blocco dei Timer

¹⁵l'overflow di un contatore rappresenta il raggiungimento della massima soglia, dopo la quale esso ripartirà dall'inizio (tipicamente, il valore zero).

La risoluzione del timer e il tempo di interruzione dipendono dal clock di ingresso selezionato e dal valore del prescaler¹⁶ impostato.

Il timer watchdog

In un sistema complesso i problemi possono verificarsi e, di fatto, si verificano; un *timer watchdog* (WDT) è una buona difesa.

Il timer watchdog è contatore che conta verso l'alto in grado di generare un impulso sul pin di reset e un interrupt ai moduli di sistema conseguentemente all'occorrere di una condizione di overflow. Il timer watchdog trasmette un segnale di reset al modulo PRCM (vedi 2.6) e trasmette una interruzione di tipo watchdog al core ARM; Il reset del modulo PRCM causa un *warm reset* del dispositivo.

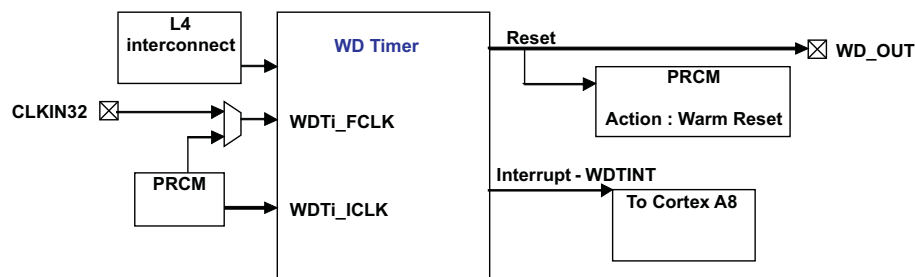


Figura 2.13: Schema a blocchi del blocco dei Timer

Un timer di questo tipo non dovrebbe mai raggiungere la condizione di overflow in un sistema funzionante: una routine di programma dovrebbe sempre resettarlo al valore iniziale, ad intervalli regolari: se ciò non accade, al timeout l'hardware resetta la CPU, riportando subito il sistema nella condizione di funzionamento normale.

Il timer watchdog di questo sistema viene temporizzato tramite il clock a 32 kHz. La prima regola che è importante seguire per il design di questo contatore è quello di connetterlo all'ingresso di reset della CPU (POR, Power On Reset), non di connetterlo ad un interrupt (come NMI, Interrupt Non Mascherabile): quando un watchdog invia il segnale di reset, significa che qualcosa di *critico* è accaduto, qualcosa che potrebbe aver lasciato la CPU in uno stato imprevedibile, quindi

¹⁶Il prescaler è un contatore elettronico utilizzato per ridurre segnali in alta frequenza in segnali a frequenza minore tramite una divisione fra interi.

solo un suo reset può riportare il sistema online. L'interrupt non mascherabile è interessante da sfruttare, il problema è che esso (e tutti gli altri interrupt) non è infallibile: su alcune CPU, una logica interna in uno stato non determinato può arrestare la risposta di questa classe di interruzioni.

2.7.2 Il bus di comunicazione I²C

La periferica I²C è un bus di tipologia multi-master: in questa implementazione, il bus fornisce un'interfaccia fra il processore ARM e qualsiasi dispositivo che sia I²C compatibile (cioè che sia connettabile sul bus seriale I²C). Tramite l'interfaccia a due fili, è possibile trasmettere (e ricevere) dati da 8-bit da (e per) la CPU. Il bus I²C è un bus multi-master: ciò significa che più di un dispositivo connesso può essere in grado di controllare il bus; ogni dispositivo I²C viene identificato da un *indirizzo univoco* e può operare sia come trasmettitore o ricevitore, a seconda della funzione necessaria.

Oltre ad essere un trasmettitore o ricevitore, un dispositivo collegato al bus I²C può anche essere considerato nell'ottica di master/slave quando si esegue un trasferimento dati: un dispositivo viene definito *master* se avvia un trasferimento di dati sul bus e *genera i segnali di clock* per permettere tale trasferimento; durante questo trasferimento, *qualsiasi dispositivo* indirizzato dal master viene considerato come uno *slave*.

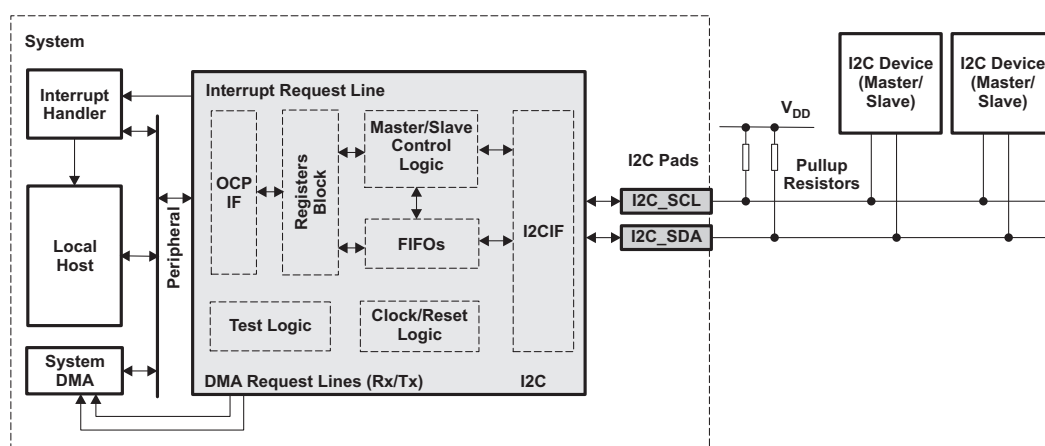


Figura 2.14: Schema a blocchi del bus I²C

Nel progetto ho utilizzato quest'ultima tipologia di connessione fra processore,

codec audio e power manager: i dati vengono comunicati ai rispettivi dispositivi tramite la linea dati seriale (SDA) e la linea di clock seriale (SCL); sia SDA e SCL sono pin bi-direzionali e **devono** essere collegati ad una tensione positiva di alimentazione tramite una resistenza di pull-up: così facendo, quando il bus è libero entrambi i pin sono settati ad un valore logico alto, poiché il driver di questi due è caratterizzato da un'uscita *open drain*; questo comporta il vantaggio di non avere in nessun caso conflitti hardware, perché nessun dispositivo può forzare il livello logico alto.

2.7.3 Il bus USB

Il controller USB fornisce una soluzione di connettività a basso costo per i dispositivi consumer portatili fornendo un meccanismo per il trasferimento di dati tra dispositivi USB con una velocità teorica massima di 480 Mbps. Il sottosistema USB è dotato di due moduli indipendenti USB 2.0 costruito su due controller OTG¹⁷. Anche se i controller sono di tipo OTG, tali peculiarità non sono supportate da nessuna delle porte: ogni porta ha il supporto per la funzionalità *dual-role* (i ruoli sono “host” e “device”) ma non è in grado di scambiare il ruolo al volo¹⁸; entrambe le porte possono operare indipendentemente l'una dall'altra.

Ogni controller USB è costruito basandosi sul controller Mentor USB OTG (musbmhdrc) e sul Synopsys SR70LX PHY[5]. Ogni controller USB possiede 32 KBytes di endpoint FIFO configurabili dall'utente ed ha il supporto per 15 endpoint di trasmissione e 15 di ricezione in aggiunta all'Endpoint 0.

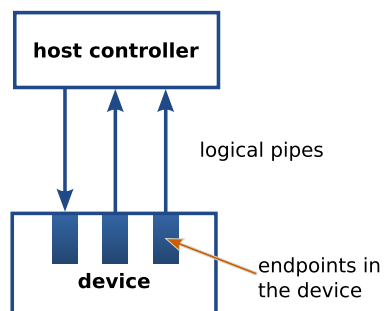


Figura 2.15: Gli endpoint USB

Ma cerchiamo di capire cos'è un endpoint: gli *endpoint* sono i punti di accesso unidirezionale per comunicare con un dispositivo; sono dei buffer preposti alla memorizzazione temporanea dei dati in entrata o in uscita dal dispositivo; all'in-

¹⁷La specifica On-The-Go permette ai dispositivi che generalmente svolgono il ruolo di USB *slave* (ad esempio un telefono cellulare) quando connessi verso un host USB (ad esempio un computer) con un altro dispositivo USB, di **diventare un USB host** se accoppiati con un altro dispositivo USB.

¹⁸come il protocollo On-The-Go prevederebbe.

terno di ogni interfaccia possono essere specificati zero o più endpoint. All'interno di una configurazione specifica, ogni endpoint ha un *indirizzo univoco*, il *numero di endpoint* e la sua *direzione*. In qualsiasi configurazione, l'endpoint predefinito (endpoint 0) non fa parte di nessuna interfaccia: è gestito dal livello dei servizi e non risulta direttamente a disposizione dei driver di periferica.

Il sottosistema USB prevede dell'hardware DMA dedicato al fine di accelerare lo spostamento dei dati; esistono due tipi di trasferimento di dati, cioè basati su *stream* e su *messaggi*. Vediamoli in dettaglio:

- *trasferimenti isocroni*, cioè avvengono ad un preciso data rate (spesso, ma non necessariamente, il più velocemente possibile), ma con possibile perdita di dati (ad esempio i flussi audio o video in tempo reale);
- *trasferimenti di tipo bulk*, cioè trasferimenti sporadici di grandi quantità utilizzando tutta la banda disponibile che rimane, ma senza alcuna garanzia né sulla larghezza di banda garantita, né sulla latenza (un esempio di ciò può essere il trasferimento di file);
- *trasferimenti ad interrupt*, specializzati per i dispositivi che devono garantire risposte rapide (cioè quella che viene definita una *bassa latenza*, come ad esempio le tastiere ed i dispositivi di puntamento come i mouse);
- *trasferimenti di controllo*, tipicamente sfruttati per inviare al dispositivo comandi brevi e semplici, ed ottenere una risposta di stato.

Il layer fisico USB Come si può notare in figura 2.16, il controller USB si interfaccia con le porte di tipo host tramite i livelli fisici. Il livello fisico (comunemente abbreviato in *PHY*) è il primo, il più complesso ed il più basso livello del modello di networking *OSI*¹⁹; è composto dalle tecnologie hardware basilari preposte alla trasmissione di rete ed è il livello fondamentale delle strutture dati logiche su cui si basano le funzioni nei livelli superiori.

Il livello fisico definisce le modalità di trasmissione di bit grezzi piuttosto che pacchetti di dati logici attraverso un link fisico che collega i nodi di rete. Il flusso

¹⁹L'*Open Systems Interconnection model* (OSI model) è un modo di suddividere un sistema di comunicazione in parti più piccole (chiamate *livelli*) proposto dalla International Organization for Standardization.

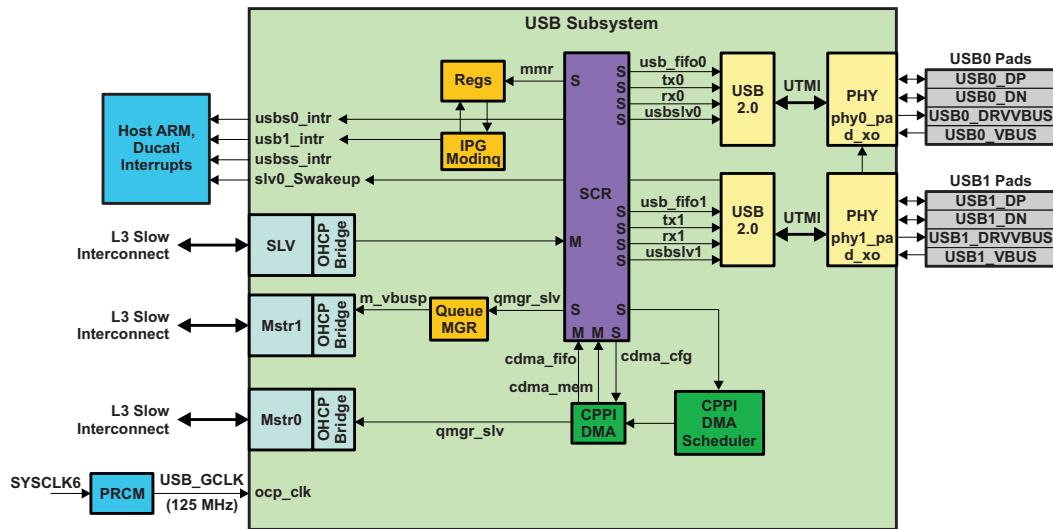


Figura 2.16: Schema a blocchi del sottosistema USB dell'AM3892

di bit può essere raggruppato in parole di codice o simboli e convertito in un segnale fisico, il quale viene trasmesso attraverso un hardware di trasmissione.

Il livello fisico fornisce un'interfaccia *elettrica, meccanica e procedurale* per il mezzo di trasmissione; son qui specificate le forme e le proprietà dei connettori elettrici, le frequenze di trasmissione, lo schema di modulazione da usare e simili parametri di basso livello. Nella semantica dell'architettura di rete OSI, il Physical Layer traduce le richieste logiche del livello *Data Link* in operazioni hardware specifiche, in grado di modellare la trasmissione o la ricezione di segnali elettronici.

Un chip PHY è integrato nella maggior parte dei sistemi embedded e fornisce il ponte di comunicazione tra la parte digitale e la parte modulata dell'interfaccia. Ogni modulo USB dispone di un layer fisico Synopsys DesignWare USB 2.0, il quale non dispone di circuito di charge pump interno, quindi, quando il sistema opera da host, è richiesta una fonte di alimentazione esterna per la tensione 5V (VBUS).

I segnali in uscita dal layer fisico sono i seguenti e sono connessi tramite linee differenziali ai receptable²⁰ USB di tipo A presenti nella scheda:

- USB_DP e USB_DN: questi due segnali rappresentano la *coppia differenziale* dedicata ai dati USB; una coppia differenziale minimizza le interferenze

²⁰ovvero ai connettori USB

elettromagnetiche, il crosstalk, sia l'emissione che l'accettazione del rumore, e consente di realizzare una caratteristica di impedenza nota, permettendo tecniche di *adattamento di impedenza* (molto importanti, ad esempio, per una linea di trasmissione ad alta velocità o per segnali audio, cioè bilanciati e ad alta qualità);

- **USB_DRVVBUS**: segnale di abilitazione dell'erogazione della tensione di alimentazione (5V); il pilotaggio di questo segnale è automatico ed è gestito dal controller USB; esso deve essere trasparente per l'utente fintantoché stanno avvenendo la connessione hardware e l'inizializzazione software; dopo questa fase, quando assume il ruolo di un host, il controller USB porta alto il segnale USB_DRVVBUS; quando invece viene assunto il ruolo di device, il controller USB porta basso il segnale USB_DRVVBUS, disabilitando la logica di potenza esterna (charge pump): di conseguenza, non vi è nessun flusso di potenza sulla linea VBUS (in questo caso, come è corretto che sia, l'alimentazione dovrebbe essere fornita dall'host esterno);
- **USB_VBUS**: tensione di alimentazione del bus USB; quando uno dei controller assume il ruolo di host, la porta USB è tenuta ad erogare una fonte di alimentazione a 5V per il dispositivo collegato: per far ciò, il controller USB richiede l'abilitazione di una logica di alimentazione esterna (o un circuito a pompa di carica) in grado di fornire la tensione richiesta.

I clock necessari Nel sottosistema USB vengono utilizzati due clock principali, il clock OCP e quello per il layer fisico. Il clock OCP (SYSCLK6 o clock di sistema), è il clock che viene utilizzato per temporizzare il controller e tutti i blocchi del sottosistema ad eccezione del PHY; SYSCLK6 è derivato dal PLL principale ed è controllato dal PRCM, il quale garantisce una frequenza di clock di 60 MHz al fine di garantire il funzionamento la larghezza di banda necessaria per avere un modo di funzionamento USB 2.0 (cioè di 480 Mb/s).

Il clock del layer fisico a 60MHz viene derivato tramite il PLL interno da un clock alla frequenza di 24 MHz fornito esternamente.

distribuzione della potenza: il power switch

I segnali differenziali e l'abilitazione dell'alimentazione vengono prodotti dal livello fisico, però questo non è sufficiente per gestire l'alimentazione delle porte USB: come accennato nel paragrafo precedente, è necessario predisporre un *circuito a pompa di carica*²¹. La mia scelta è ricaduta sul TPS2052B prodotto da Texas Instruments, poiché è in grado di fornire due uscite completamente indipendenti (quindi può erogare l'alimentazione a due porte USB in configurazione *host*) ed eroga una corrente di 500 mA per porta (che è il valore stabilito dallo standard USB).

Lo switch di distribuzione TPS2052B incorpora due switch di potenza a mosfet di tipo *n*, caratterizzati da una R_{DSon} di 70 m Ω . Ogni switch è controllato da un segnale logico di abilitazione (USB_DRVVBUS). Il pilotaggio dei gate dei mosfet è gestito da un "charge pump" progettato per controllare i tempi di salita e di discesa, con lo scopo di ridurre al minimo i picchi di corrente durante le commutazioni.

Il dispositivo è dotato di un *sistema di protezione*, il quale limita la corrente di uscita ad un livello tollerabile passando alla modalità *a corrente costante* quando la corrente di carico supera la soglia limite (o è presente un cortocircuito); quando la protezione si innesca, lo segnala portando il pin OC ad un valore logico basso. E' inoltre presente una protezione termica del dispositivo, incaricata di disattivarlo se la temperatura operativa esce dal range imposto dal costruttore; tale circuito è automatico e ri-abilita automaticamente il dispositivo, una volta che le condizioni operative ritornano sicure.

2.7.4 Il sistema audio

Il sistema audio scelto per il progetto consta di due distinti componenti: l'interfaccia audio interna al processore AM389x (McASP) e il codec audio TLV320-AIC3106²²; nell'accezione hardware del termine, si dice *codec audio* un dispositivo che opera la codifica dell'audio analogico in opportuni segnali digitali ed opera la decodifica di segnali digitali in segnali analogici: esso contiene sia un ADC

²¹Circuito a pompa di carica: è una sorta di convertitore DC-DC che utilizza dei condensatori come elementi di stoccaggio dell'energia, per creare una fonte di maggiore o minore tensione.

²²fornito da Texas Instruments.

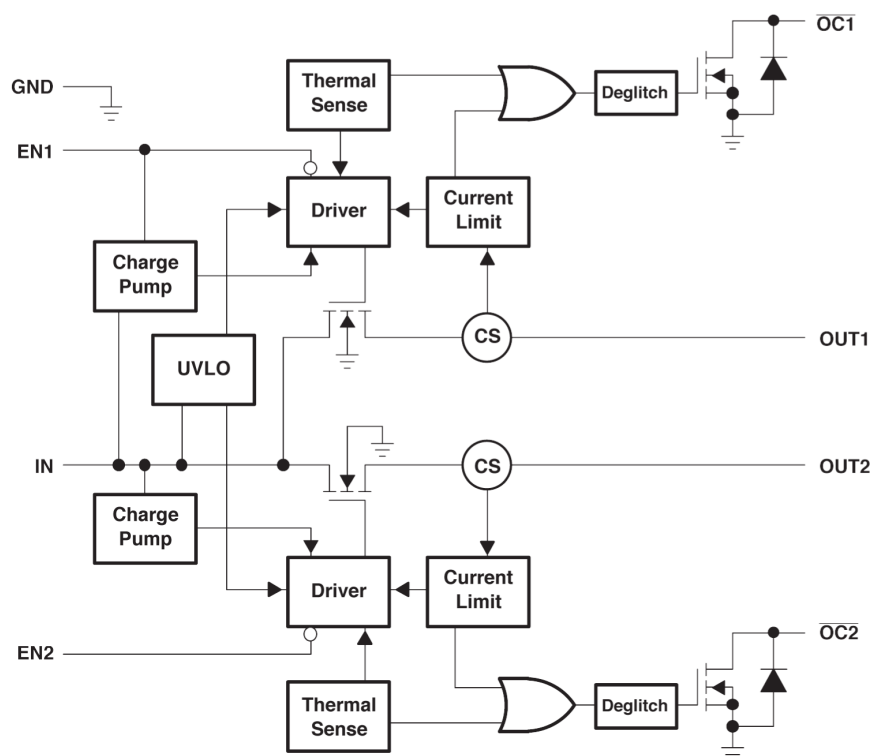


Figura 2.17: Schema a blocchi del charge pump TPS2052B

che un DAC, pilotati dallo stesso clock. L'interfaccia audio multicanale seriale (denominata "McASP" per la cpu AM3892 di Texas Instruments) opera come una porta seriale audio general-purpose, ottimizzata per le esigenze delle applicazioni audio multicanale; è utilizzabile per flussi time-division multiplexing (TDM), per il protocollo Inter-Integrated Sound (I²S), e per la trasmissione di tipo DIT (Digital audio Interface Transmission).

Come si nota dalla figura 2.18, McASP è costituito da sezioni di trasmissione e ricezione che possono operare sincronizzate oppure in piena autonomia (grazie a master clock, bit clock e frame sync separati) ed essere utilizzate con diversi metodi di trasmissione e con diversi formati di bit-stream; McASP comprende 6 serializzatori che possono essere attivati individualmente per trasmettere o per ricevere.

Il formato I²S I²S rappresenta uno standard elettrico di un'interfaccia seriale utilizzata per il collegamento di dispositivi audio di tipo digitale; è comunemente

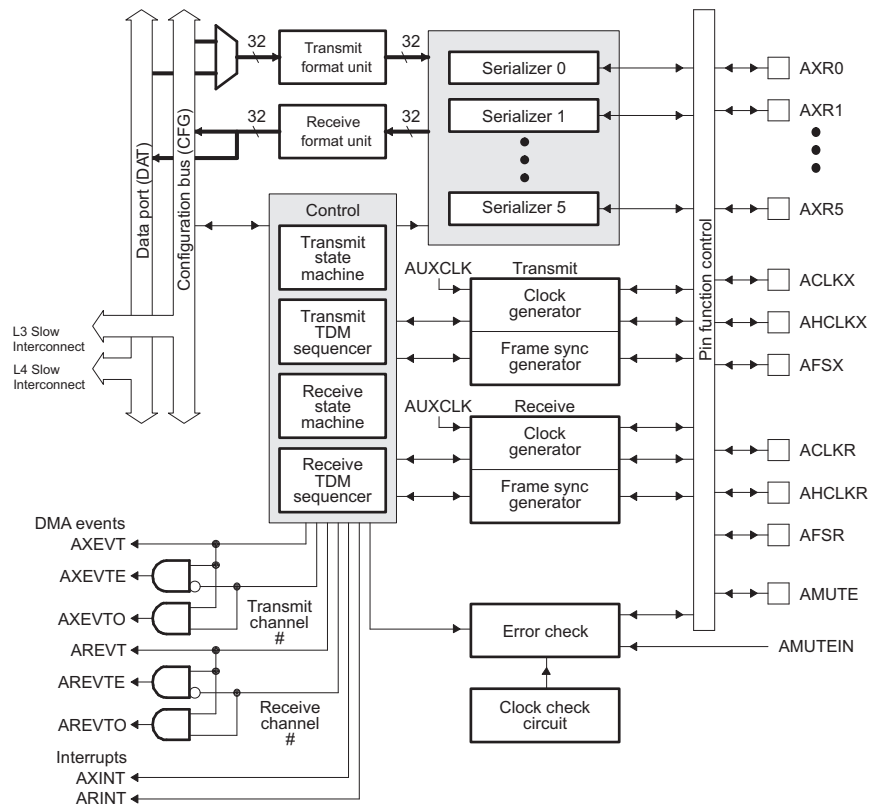


Figura 2.18: Schema a blocchi del sottosistema audio dell'AM3892

utilizzata per trasportare informazioni di tipo PCM fra CD²³ e il DAC²⁴ presente in un lettore CD; per come è definito lo standard, il bus I²S separa i segnali clock da quelli dei dati: la connessione ha quindi un *jitter* molto basso; per definizione, il jitter è la *deviazione o spostamento* di alcuni aspetti degli impulsi in alta frequenza del segnale digitale: come suggerisce il nome, esso può essere pensato come impulsi instabili; la deviazione può essere in termini di ampiezza, i tempi di fase, o la larghezza dell'impulso di segnale.

Il coded audio

Nella board di valutazione offerta dal Texas è presente il codec audio TLV320-AIC3106[6]: è un codec stereofonico a basso consumo di potenza, dotato di un amplificatore per cuffie stereo e di più ingressi e più uscite programmabili in configurazione single-ended oppure completamente differenziale; questo codec è

²³Compact Discs.

²⁴sta per "Digital-to-Analog Converter", cioè un convertitore digitale-analogico.

2 PROGETTAZIONE DEL SISTEMA

comodo per il firmwarista poiché la gestione del flusso di potenza è controllabile in modo estensivo tramite dei registri²⁵. Il TLV320AIC3106 contiene quattro driver di uscita ad alta potenza e tre driver completamente differenziali: i primi sono in grado di pilotare una vasta gamma di configurazioni di carico, tra cui fino a quattro canali di cuffie a 16 Ω single-ended usando condensatori per accoppiamento in alternata (AC-coupling) oppure in configurazione capacitorless.

Al TLV320AIC3106 viene fornita una tensione di alimentazione analogica di 3.3V, una tensione di core digitale di 1.8V ed una tensione digitale di 3.3V per la logica di input-output.

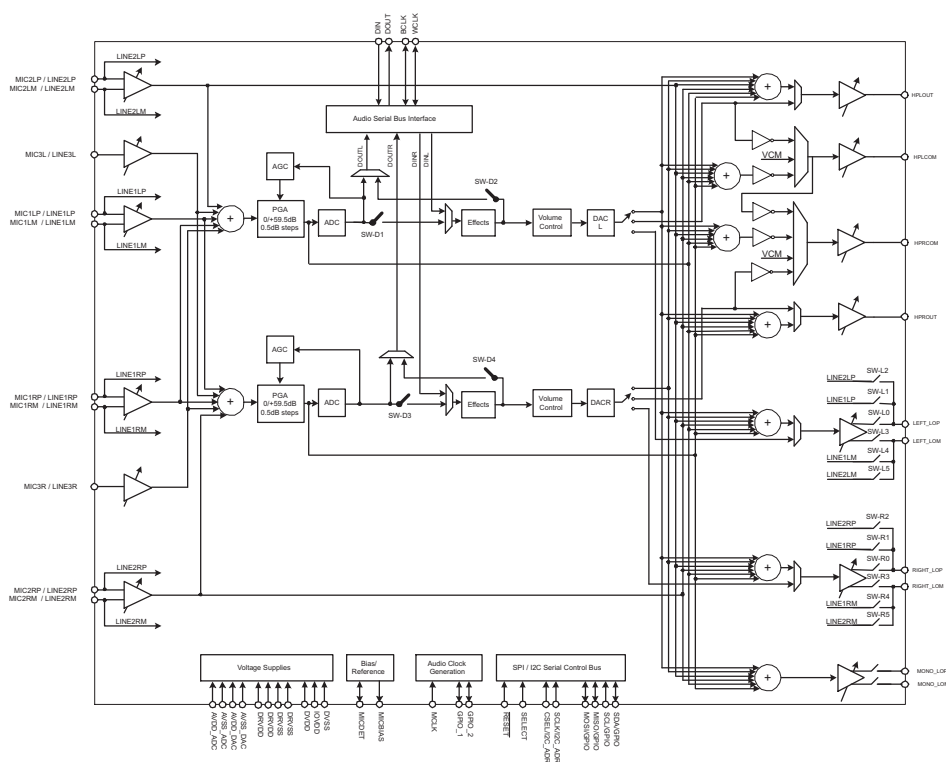


Figura 2.19: Schema a blocchi del codec audio TLV320AIC3106

²⁵Un *registro hardware* memorizza i bit di informazione, in modo che tutti i bit possano essere letti o scritti in contemporanea. La lettura di registri hardware residenti nelle unità periferiche, in generale tutti i componenti hardware esterni alla CPU, comportano che l'accesso viene eseguito attraverso una tecnica di indirizzamento *memory-mapped*, mappato su indirizzi di memoria, oppure *port-mapped*, mappato su indirizzi riservati a porte I/O; l'accesso avviene con istruzioni di caricamento oppure di memorizzazione attivate dal processore.

I clock del codec Il codec viene fornito con all'interno un PLL altamente programmabile, necessario per generare tutti i clock e supportare tutte le frequenze previste dagli standard audio; vi è una vasta gamma di MCLKs disponibili, che variano da 512 kHz a 50 MHz, comprendendo quindi i clock più comunemente utilizzati (12 MHz, 13 MHz, 16MHz, 19.2 MHz, e 19.68 MHz).

La configurazione Il codec audio in esame (presente sulla EVB) è più che sufficiente per le esigenze audio del sistema finale, quindi è stato selezionato per essere equipaggiato nel sistema embedded; la scelta è stata compiuta tenendo conto sì delle caratteristiche hardware ma anche del fatto che sono già disponibili i device drivers per Android: questa caratteristica consente di sgravare il firmwarista dal compito di doverli progettare.

Viene qui illustrata (si veda la figura 2.20) la configurazione consigliata di interconnessione del codec; per realizzare la soluzione ottima per il sistema oggetto di tesi sono partito da questa, adattandola alle necessità specifiche: ho provveduto alla rimozione dei condensatori di disaccoppiamento per le uscite che non verranno sfruttate e alla semplificazione dei condensatori di livellamento della tensione, nonché quelli di filtro.

2.7.5 Lo slot di espansione SecureDigital

Molti sistemi embedded sfruttano i supporti SecureDigital come principale meccanismo di memorizzazione dei dati, spesso affiancandosi a dei controllori: infatti AM3892 contiene un controllore di tipo *SecureDigital* in grado di fornire un'interfaccia tra microprocessore (MPU) e le schede di memoria SD. Il controllore si occupa di dialogare con la scheda SD a livello di trasmissione, di pacchettizzazione dei dati, di aggiunta dei controlli a ridondanza ciclica (CRC), di inserimento dei bit di inizio/fine, e di controllo per la correttezza sintattica.

Il controllore è in grado di inviare ogni comando SD, effettuare un polling per conoscere lo stato della scheda oppure attendere una richiesta di interruzione, la quale viene reinviata indietro in occorrenza di eccezioni (o segnalare il termine di un'operazione).

Il controller, infine, supporta anche l'accesso diretto alla memoria, tramite due canali DMA.

2 PROGETTAZIONE DEL SISTEMA

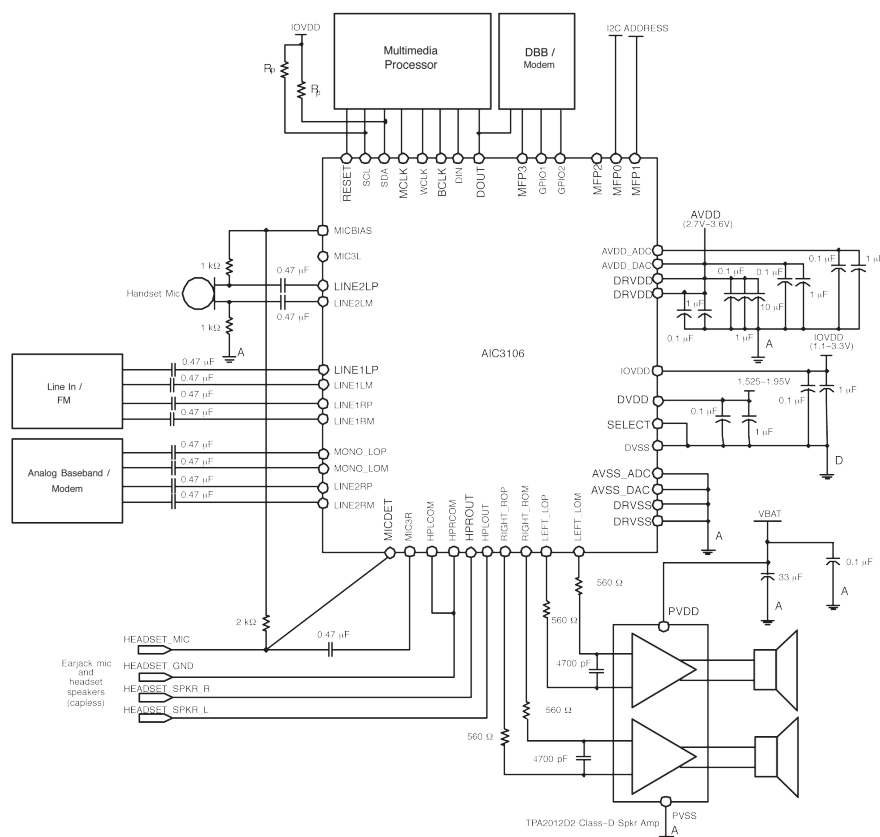


Figura 2.20: Configurazione tipica del codec audio TLV320AIC3106

La specifica di progetto richiedeva l'inclusione di uno slot μSD : vediamo come l'ho realizzato, studiandone i segnali interessati che il controllore rende disponibili (vedi figura 2.21):

- SD_CLK: questo pin fornisce alla scheda SD il segnale di clock;
- SD_CMD: questo pin viene impiegato per offrire una comunicazione bidirezionale fra controllore e scheda SD: il controllore invia i comandi alla scheda, la scheda restituisce la risposta ai comandi ricevuti;
- SD_DAT[3:0]: questo bus rappresenta il bus dati fra controllore e scheda;
- SD_POW: questo pin è preposto all'abilitazione dell'alimentazione verso la scheda SD: quando è a un valore logico alto significa che c'è alimentazione;
- SD_SDCD: questo pin ha il compito di fungere da *carrier detect*; il segnale viene comandato da uno switch meccanico presente nello slot;

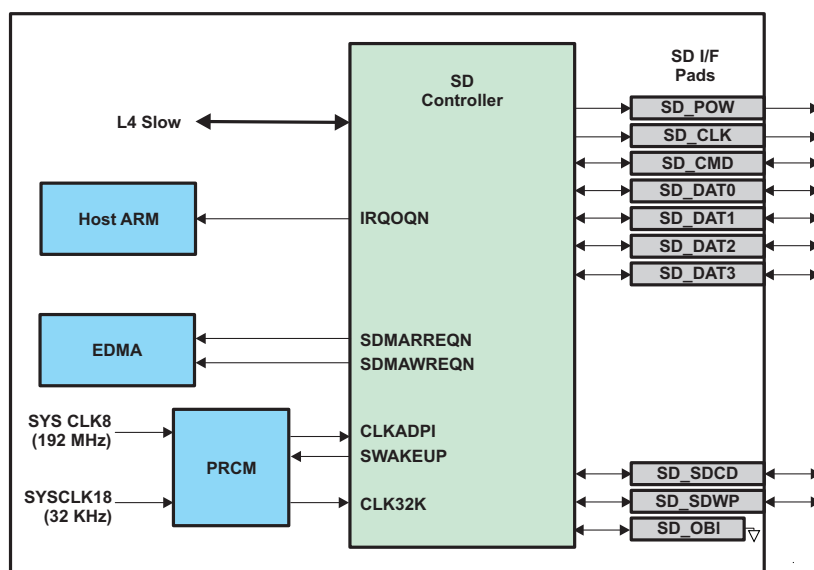


Figura 2.21: Schema del controllore SecureDigital

- **SD_SDWP:** questo pin serve per proteggere da scrittura la scheda SD; anche questo segnale è generato da uno switch meccanico presente sulla scheda SD.

Protocollo e formato dei dati Il protocollo di comunicazione fra il controllore e la scheda SD è basato su messaggi; ogni messaggio è composto da una delle seguenti parti:

- *Comando:* un comando consente di avviare un'operazione e viene trasferito serialmente dal controller alla scheda;
- *Risposta:* una risposta è una risposta in seguito ad un dato comando; la risposta viene inviata serialmente dalla scheda SD al controller SD;
- *Dati:* i dati vengono trasferiti dal controller alla scheda utilizzando le linee dati;
- *Occupato:* il segnale SD_DAT0 è mantenuto basso dalla scheda finché essa sta elaborando i dati ricevuti;
- *Status CRC:* il risultato del controllo di ridondanza ciclica viene inviato dalla scheda attraverso la linea SD_DAT0 durante l'esecuzione di una scrittura.

ra: in caso di errore di trasmissione (che si può verificare su di una delle righe di dati attive), la scheda invia uno status di CRC negativo su SD_DAT0. In caso di successo su tutte le linee di dati attive, la scheda invia uno stato di CRC positivo su SD_DAT0 e avvia la procedura di programmazione dei dati.

Per quanto riguarda il formato dei dati, l'interfaccia SecureDigital supporta solo operazioni basate su lettura e scrittura a blocchi dei dati.

2.7.6 La porta RS-232

Il sistema embedded è corredato di una porta²⁶ di tipo UART²⁷ RS-232, la quale permette un sistema di scambio dei dati a bassa velocità tra dispositivi digitali; questa tipologia di interconnessione è da ritenersi obsoleta per sistemi moderni general-purpose (di fatto, è stata soppiantata dal bus USB), ma è anche una buona connessione per il debug di sistemi embedded; di fatto, ogni famiglia di microprocessori prevede questa interfaccia di comunicazione.

La UART esegue conversioni seriale — parallelo sui dati ricevuti da una periferica e la conversione parallelo — seriale sui dati ricevuti dalla CPU.

Durante la configurazione del sistema di test Android non mi sono avvalso di dell'interfaccia grafica per nessuna delle Evaluation Boards, bensì ho utilizzato un terminale da riga di comando, connesso alla scheda tramite *Minicom*: Minicom è un modem controller ed un emulatore di terminale²⁸ disponibile per i sistemi operativi *nix; un uso comune per questo applicativo è la creazione di un terminale seriale remoto, spesso l'ultima risorsa per accedere a un computer se non è disponibile una connessione di tipo LAN, oppure al momento di boot del sistema. I trasferimenti su interconnessione UART sono di tipo *asincrono*: asincrono significa, in questo contesto, che i dati vengono trasmessi, byte per byte, in modo anche non consecutivo e senza l'aggiunta di un segnale di clock, cioè di un segnale comune che permette di sincronizzare la trasmissione con la ricezione; ovviamente sia il trasmettitore che il ricevitore devono comunque essere dotati di un clock

²⁶In realtà il core AM3892 prevede fino a tre interfacce di tipo UART (dipende da come vengono multiplexati i pin), ma la richiesta di progetto è per una sola interfaccia.

²⁷sta per “Universal Asynchronous Receiver-Transmitter”, ovvero “ricevitore-trasmettitore asincrono universale”.

²⁸Minicom supporta le emulazioni ANSI e VT100.

locale per poter interpretare i dati; la sincronizzazione dei due clock è comunque necessaria ed è fatta in corrispondenza della prima transizione sulla linea dei dati.

Lo standard EIA RS-232 Lo standard EIA RS-232 prevede fino a 20 segnali da definire, ma offre completa libertà per l'utente; sono sufficienti tre fili: uno per inviare i dati (TX), uno per la ricezione dei dati (RX) e la massa di segnale (GND); le linee rimanenti possono essere cablate o disattivate in modo permanente. La trasmissione del segnale è bipolare e richiede due tensioni, da 5 a 25 volt, di polarità opposta. Lo standard industriale prevede di utilizzare una parola asincrona composta da un bit di start, sette o otto bit di dati, un bit di parità opzionale ed uno o due bit di stop.

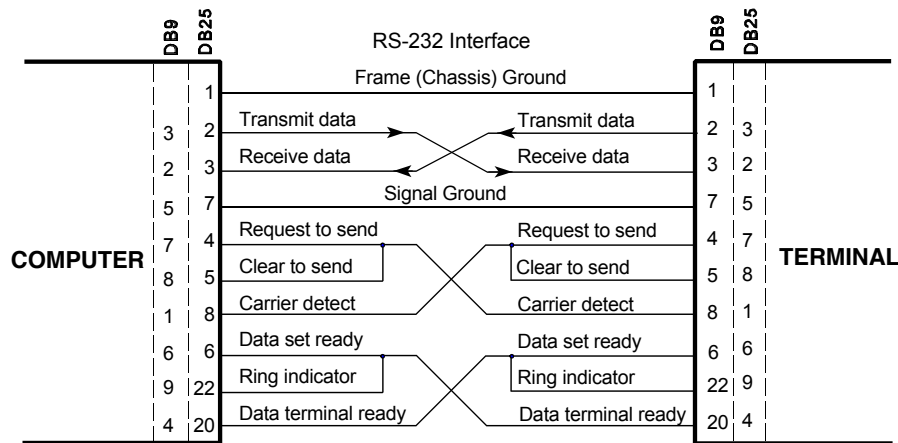


Figura 2.22: RS-232: connessione punto-a-punto

Come possiamo vedere dalla figura 2.22, molti dei segnali previsti per i modem non sono necessari se stiamo connettendo punto a punto due computer (è il nostro caso, connettiamo un computer di test ed il dispositivo embedded).

L'standard RS-232 specifica che il valore logico "1" deve essere inviato come tensione nella gamma da -15 a -5 V e che il valore logico "0" deve essere inviato come tensione nella gamma da +5 a +15 V. La norma specifica che le tensioni di almeno 3 V di ampiezza (vedi figura 2.23) devono essere sempre riconosciute correttamente verso dal ricevitore in base alla loro polarità, affinché possa essere tollerata un pò di attenuazione lungo la linea.

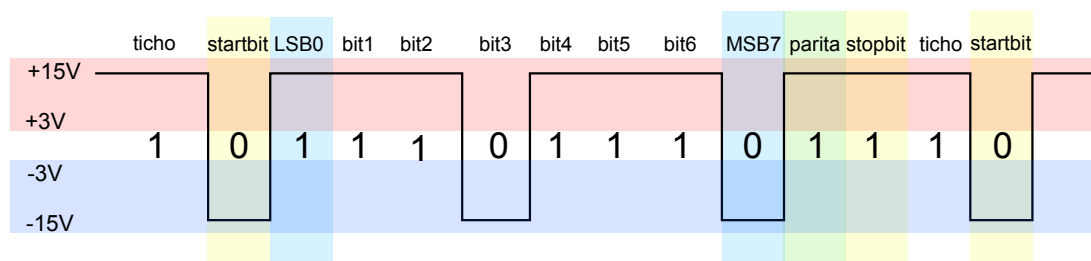


Figura 2.23: Un esempio di trasmissione tramite RS-232

La velocità di trasferimento è stimata come maggiore di 20 kbps ad una distanza minore di 15 m; con un buon design è possibile raggiungere anche distanze superiori, ma è ragionevole utilizzare questo limite dato come limite superiore. L'impedenza di carico al ricevitore deve essere in un valore compreso fra 3 e 7 $k\Omega$ e non più di 2.5 μF .

2.7.7 Il sistema video

L'output video richiesto da specifica deve essere di tipo LVDS: vediamo cos'è questa tipologia di segnali e come è progettata, per meglio capire come andrà realizzato poi il circuito.

Lo standard LVDS Lo standard LVDS (*Low Voltage Differential Signaling*) è stato creato per quelle applicazioni in cui le comunicazioni dei dati devono avvenire ad una elevata velocità di trasferimento; è una soluzione a basso costo, definita nella norma TIA/EIA-644: si tratta di una linea di trasmissione a basso voltaggio, bassa potenza, utilizzata principalmente per le applicazioni di tipo punto—punto e multi—drop; la massima velocità di trasmissione dei dati è di 655 Mbps, anche se alcune delle applicazioni odierne riescono a spingersi addirittura oltre.

come si nota dalla figura 2.24, rispetto al link differenziale RS422 e RS485, LVDS è caratterizzato da uno swing differenziale di tensione più basso (tipicamente 350 mV) con una tensione di offset tipica di 1.25V rispetto al riferimento; ciò consente, ad esempio, di ottenere una buona reiezione al rumore di modo comune (Common-Mode Reject Ratio, CMRR). Infatti, un basso swing differenziale di segnale può anche mitigare problematiche di integrità del segnale alle alte velocità. Vediamo un altro esempio per cui LVDS è conveniente: man mano che

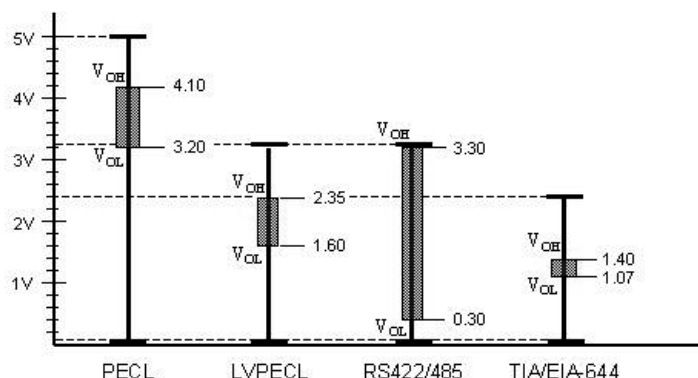


Figura 2.24: Confronto fra LVDS ed altri tipi di segnale noti

la linea subisce un aumento del carico, *l'impedenza caratteristica può cambiare* e provocare squilibri che generano *riflessioni* su tutta la linea di trasmissione; tali riflessioni possono causare errori nei bit o aumentare i setting times: LVDS risolve questo problema accettando il rumore di modo comune sulla linea differenziale e generando intrinsecamente meno disturbo, a causa del basso swing di tensione.

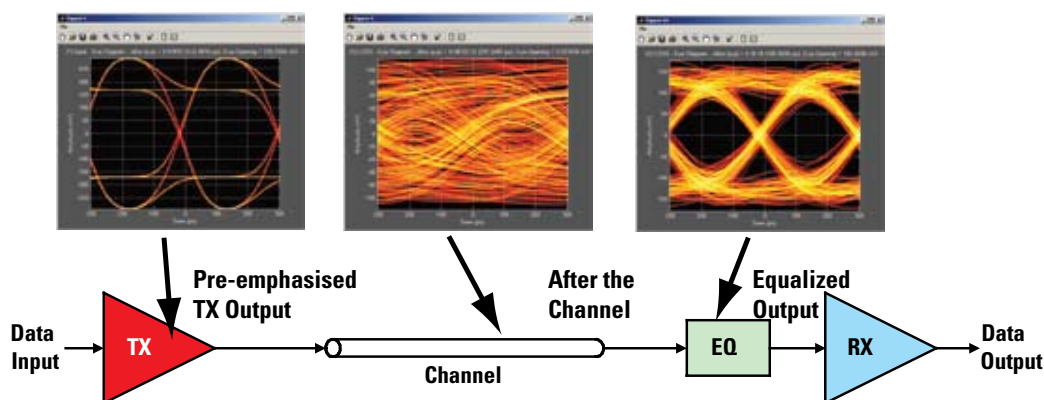


Figura 2.25: Rappresentazione tipica del segnale LVDS nel canale

Lo standard LVDS necessita di *terminazione* della linea di trasmissione: essa avviene all'ingresso del ricevitore al fine di generare l'output di tensione differenziale; la norma TIA / EIA-644 specifica il valore della resistenza di terminazione interna tra i 90 Ω e 132 Ω ; nel progetto sfrutterò dei valori di resistenza per le uscite pari a 100 Ω , poiché i produttori consigliano di rimanere all'interno del range 90—110 Ω .

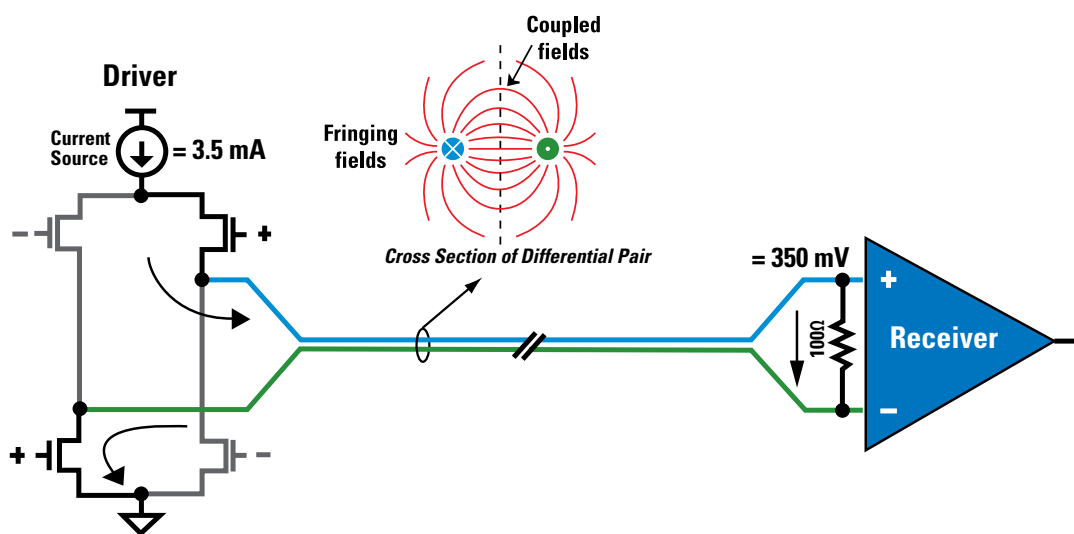


Figura 2.26: Schematizzazione di un Driver ed un Receiver LVDS

Come detto precedentemente al paragrafo 2.3.3, il core ARM esporta un segnale RGB su due uscite digitali, una a 30/24/16 bit (VOUT[0]) ed una a 16 bit (VOUT[1]), utilizzabili ad una frequenza massima di 165 MHz. Il display che farà parte del sistema possiede un'interfaccia LVDS a 18 bit: il vincolo di progetto impone di utilizzare l'uscita VOUT[0], sfruttando 6 bit per ognuno dei canali.

Le uscite video digitali sono fornite dallo High-Definition Video Processing Subsystem (HDVPSS), una periferica interna al processore, la quale fornisce una interfaccia di *ingresso video*²⁹ per periferiche esterne di imaging (sensori di immagini, decoder video,...), un'interfaccia di uscita video per i dispositivi di visualizzazione, come display SD e HD³⁰, pannelli digitali LCD, ed infine codificatori SD ed HD e un'interfaccia HDMI.

HDVPSS fornisce ai sistemi di tipo embedded anche i segnali di sincronia necessari (VSYNC, HYNC, ENABLE).

Il trasmettitore LVDS

Dato che il processore non esporta il segnale desiderato, per rimanere in specifica è necessario crearlo a partire dal segnale parallelo RGB che viene fornito: il circuito scelto per assolvere questo compito è SN65LVDS93[7], prodotto da Texas

²⁹Nel progetto non è richiesta, quindi non verrà utilizzata.

³⁰Cioè in Standard Definition e in High Definition.

Instruments.

Il SerDes³¹ contiene al suo interno quattro registri a scorrimento che accettano in ingresso 7 bit in parallelo ciascuno e li trasferiscono serialmente all'uscita, un generatore per i clock opportuni necessari e cinque driver LVDS.

Questo circuito consente quindi di inviare 28 bit di dati in logica LVTTTL single-ended su cinque linee differenziali bilanciate. Durante la trasmissione dei dati, i bit D0—D27 vengono caricati nei registri sul fronte del segnale di clock in ingresso (CLKIN)³². Il clock CLKIN viene moltiplicato per sette volte e poi utilizzato per scaricare i registri in modo seriale, sette bit per volta; successivamente, vengono inviati ai driver LVDS i quattro stream di dati ed un clock (CLKOUT) generato da un PLL alla stessa frequenza di CLKIN.

Il SN65LVDS93 non richiede particolari componenti esterni di supporto e praticamente nessun controllo: l'unica configurazione che è necessario svolgere è la selezione della polarità del clock, portando il valore di CLKSEL alto per campionare il dato sul fronte di salita e viceversa; è inoltre presente un pin di arresto (SHTDN), cioè un ingresso attivo basso in grado di disattivare il clock ed i driver LVDS di uscita per un basso consumo energetico: se settato basso, questo segnale cancella tutti i registri interni.

Riporto ora due figure (rispettivamente la 2.28(a) e la 2.28(b)) che illustrano come è costruito (mediante uno schema a blocchi) il ricevitore montato sul display[8] e come i bit RGB (ed i segnali di sincronia) vengono suddivisi nella linea LVDS e come vengono interpretati dal ricevitore: è da notare che la mappatura è sì logica, ma non prettamente intuitiva.

³¹Ovvero il Serializzatore/Deserializzatore.

³²La polarità del clock può essere selezionata scegliendo il valore di tensione alto o basso al pin CLKSEL.

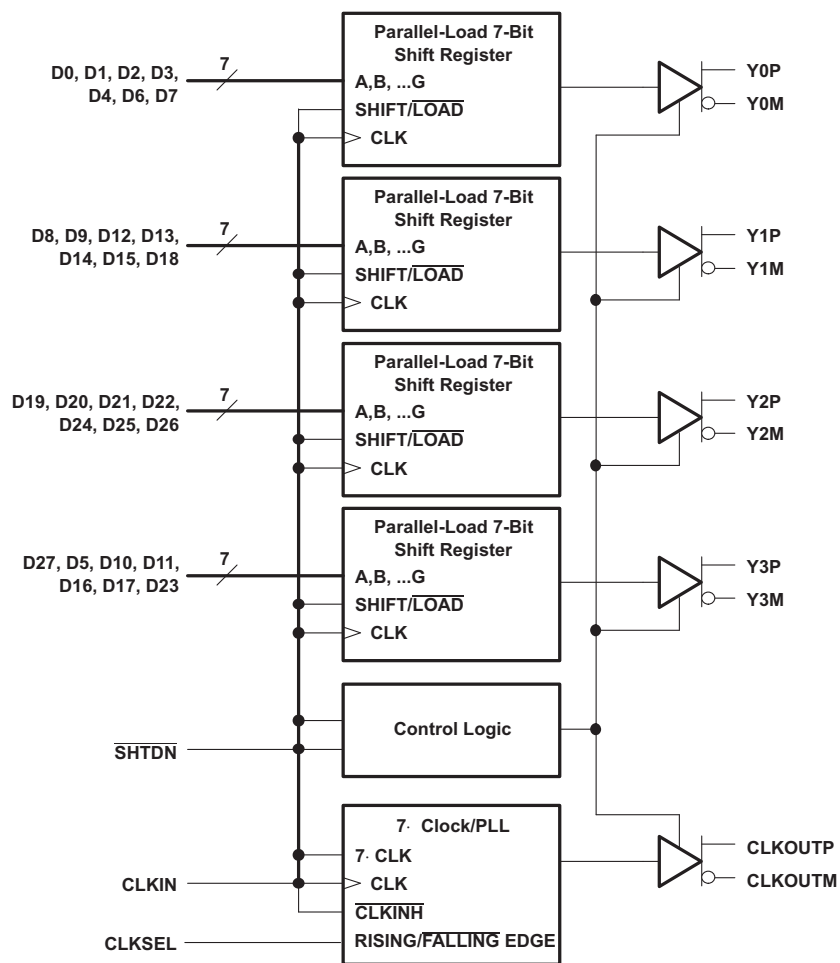
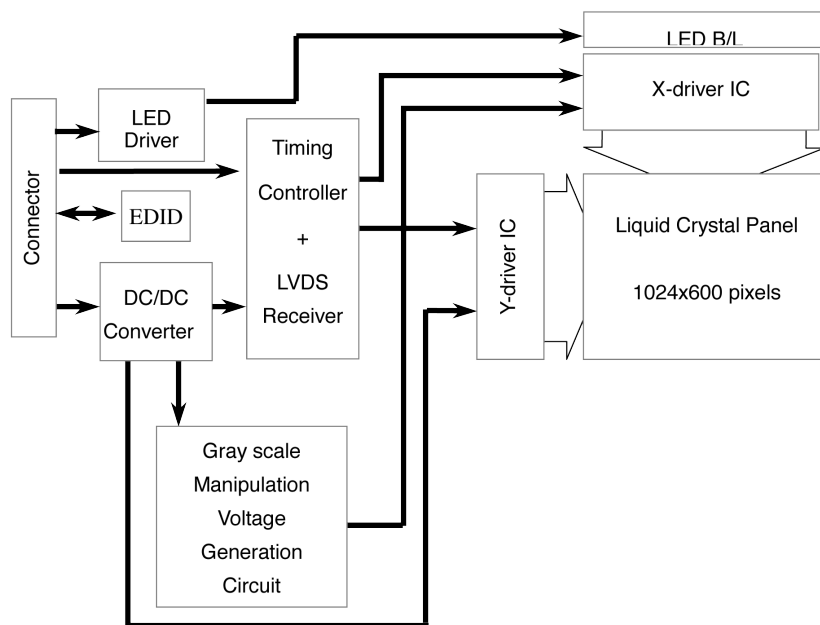
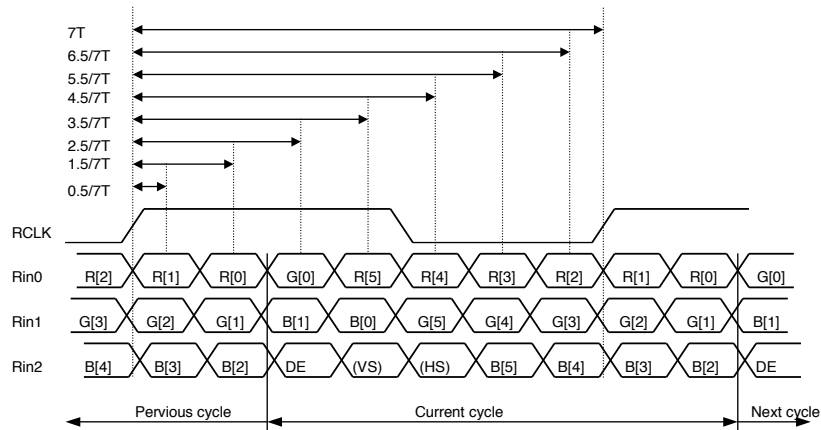


Figura 2.27: Schema a blocchi del SERDES LVDS



(a) Schema a blocchi del ricevitore LVDS



(b) Mappatura dei bit RGB e dei segnali di sincronia

Figura 2.28: Il ricevitore montato sul display touchscreen

Capitolo 3

Dimensionamento dello stadio di alimentazione

Continuando con la descrizione delle sezioni che compongono il generatore, in questo capitolo è esposta la progettazione della sezione di alimentazione. Sono descritte le tecniche adottate per la riduzione del rumore introdotto dagli stadi di alimentazione nonché le problematiche relative alla realizzazione fisica del circuito e le tecniche impiegate per ottenere un buon layout.

3.1 I convertitori a commutazione (SMPS)

I convertitori dc-dc sono molto utilizzati negli alimentatori stabilizzati a corrente continua e negli azionamenti per motori in corrente continua; spesso l'ingresso di tali convertitori è una tensione continua non regolata, la quale è ottenuta rad-drizzando la tensione di linea e che, pertanto, manifesta delle oscillazioni dovute alle variazioni di ampiezza della tensione di linea stessa.

I convertitori dc-dc a commutazione¹ vengono impiegati per trasformare la tensione continua non regolata di ingresso in una tensione d'uscita regolata al livello desiderato; le tipologie più utilizzate sono le seguenti[9]:

- convertitore *abbassatore* (*buck* o *step-down*);
- convertitore *elevatore* (*boost* o *step-up*);
- convertitore *abbassatore/elevatore* (*buck-boost*);
- convertitore *Cùk*;
- convertitore *Sepic*;
- convertitore *a ponte/mezzo ponte*.

Nei convertitori dc-dc, la tensione media di uscita deve essere controllata in modo da uguagliare il livello desiderato, anche se la tensione di ingresso e il carico di uscita oscillano; la tensione media di uscita è controllata regolando i tempi di apertura e chiusura dell'interruttore (t_{on} e t_{off}).

Il valore medio V_o della tensione di uscita v_o dipende da t_{on} e t_{off} ; uno dei metodi per controllare la tensione di uscita impiega una frequenza di commutazione costante (per cui si ha un tempo di commutazione costante $T_s = t_{on} + t_{off}$) e modifica la durata della chiusura dell'interruttore. In questa tecnica, detta *modulazione a larghezza di impulso*², viene variato il fattore di utilizzazione (normalmente indicato con il nome di *duty cycle*) D , definito come il rapporto tra il tempo di chiusura ed il periodo di commutazione ($\frac{t_{on}}{T_s}$).

Il segnale di comando dell'interruttore, che regola il suo stato di chiuso o aperto, viene generato confrontando una tensione di controllo $v_{controllo}$ con un'onda ripetitiva (sinusoide, triangolare, dente di sega): la frequenza di tale onda determina la frequenza di commutazione f_s ³. Quando il segnale di errore, nella supposizione che esso vari molto lentamente rispetto alla frequenza di commutazione, è più grande della portante triangolare, il segnale che comanda l'interruttore diventa

¹SMPS: Switched Mode Power Supply.

²PWM, Pulse Width Modulation

³nei controlli PWM essa varia tra pochi kHz ad alcune centinaia di kHz.

alto e ne determina la chiusura; altrimenti, rimane aperto. Vale quindi la seguente relazione:

$$D = \frac{t_{on}}{T_s} = \frac{v_{controllo}}{\hat{V}_{st}} \quad (3.1)$$

3.1.1 Il convertitore abbassatore di tensione

Il convertitore abbassatore fornisce al carico una tensione media di uscita più bassa di quella d'ingresso V_i . La tensione media di uscita viene calcolata in funzione del duty cycle:

$$V_o = \frac{1}{T_s} \int_0^{T_s} v_o(t) dt = \frac{1}{T_s} \left(\int_0^{t_{on}} V_d dt + \int_{t_{on}}^{T_s} 0 dt \right) = \frac{t_{on}}{T_s} V_d = DV_d \quad (3.2)$$

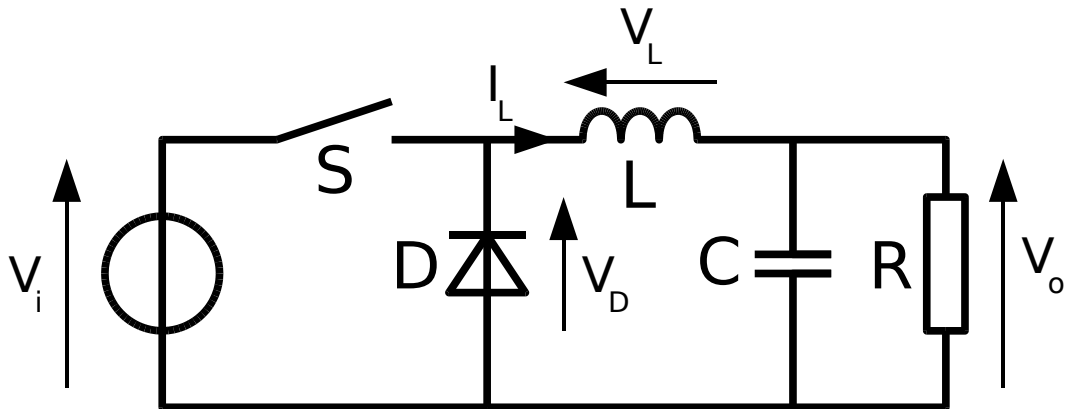


Figura 3.1: Schema elettrico semplificato di un convertitore buck

Otteniamo la relazione di controllo della tensione di uscita mediante controllo diretto del duty cycle sostituendo la (3.1) nella (3.2):

$$V_o = \frac{V_d}{\hat{V}_{st}} v_{controllo} = k v_{controllo}$$

La modalità a conduzione continua (CCM) Un convertitore opera in modalità di conduzione continua se la corrente nell'induttore circola senza discontinuità ($i_L(t) > 0$); quando l'interruttore è chiuso per il tempo t_{on} in esso circola

la corrente dell'induttore e il diodo è polarizzato inversamente ciò comporta una caduta di tensione positiva sull'induttore $v_L = V_i - V_o$ e, di conseguenza, un aumento lineare della corrente i_L . Quando l'interruttore è aperto, la corrente i_L continua a circolare grazie all'energia immagazzinata nell'induttore, che ora circola attraverso il diodo, poiché è polarizzato direttamente; si ha che la tensione sull'induttore risulta: $v_L = -V_o$.

Il comportamento appena illustrato è valido per ogni periodo in regime permanente (escludendo quindi i transitori di accensione e spegnimento del convertitore): ciò significa che, nell'intervallo-periodo T_s deve valere che l'integrale della tensione dell'induttore sia uguale a zero:

$$\int_0^{T_s} v_L(t) dt = \int_0^{t_{on}} v_L dt + \int_{t_{on}}^{T_s} v_L dt = 0 \quad (3.3)$$

Ciò significa inoltre che le aree sottese nei due sottointervalli del periodo T_s devono essere uguali, cioè:

$$(V_i - V_o) = V_o (T_s - T_{on})$$

quindi:

$$\frac{V_o}{V_i} = \frac{t_{on}}{T_s} = D \text{ (duty-cycle)} \quad (3.4)$$

Se trascuriamo le perdite associate a tutti gli elementi del circuito, si ha che il convertitore è *trasparente in potenza*, cioè la potenza d'ingresso è uguale a quella d'uscita:

$$P_i = P_o$$

Quindi,

$$V_i I_i = V_o I_o \text{ e } \frac{I_o}{I_i} = \frac{V_i}{V_o} = \frac{1}{D}$$

Si osservi che la forma d'onda della corrente istantanea in ingresso passa da un valore massimo a zero ogni volta che l'interruttore viene aperto: risulta quindi necessario prevedere un filtro appropriato, al fine di eliminare gli effetti non desiderati delle *armoniche di corrente*.

L'ondulazione della tensione di uscita (ripple) Nell'analisi eseguita precedentemente, il valore della capacità di uscita è stato stimato talmente elevato

da avere $v_o(t) = V_o$. Tuttavia, il ripple della tensione d'uscita di un valore reale di capacità può essere calcolato, in CCM, con un procedimento simile a quello svolto precedentemente sulla tensione dell'induttore: supponendo che tutto il ripple della corrente i_L circoli nel condensatore e che il valore medio di i_L circoli nel carico resistivo, l'area sottesa fra le due curve rappresenta una quantità di carica addizionale ΔQ ; la tensione di ripple picco-picco ΔV_o può essere quindi scritta così:

$$\Delta V_o = \frac{\Delta Q}{C} = \frac{1}{C} \frac{1}{2} \frac{\Delta I_L T_s}{2} \quad (3.5)$$

Durante il t_{off} si ha:

$$\Delta I_L = \frac{V_o}{L} (1 - D) T_s \quad (3.6)$$

Sostituendo la relazione (3.6) nella (3.5) si ottiene:

$$\frac{\Delta V_o}{V_o} = \frac{1}{8} \frac{T_s^2 (1 - D)}{LC} = \frac{\pi^2}{2} (1 - D) \left(\frac{f_c}{f_s} \right)^2 \quad (3.7)$$

dove $f_s = \frac{1}{T_s}$ e

$$f_c = \frac{1}{2\pi\sqrt{LC}} \quad (3.8)$$

L'equazione 3.7 mostra che il ripple della tensione di uscita può essere minimizzato progettando la frequenza di taglio f_c del filtro passa-basso in modo che sia $f_c \ll f_s$.

3.1.2 I regolatori di tensione lineari (LDO)

Il regolatore lineare rappresenta la struttura base di quasi ogni alimentazione impiegata in elettronica: nella sua forma integrata è semplice da usare ed il suo basso costo lo rende uno dei più economici componenti presenti in un sistema embedded. Il regolatore lineare fornisce una tensione in uscita costante partendo da una tensione di ingresso ugualmente costante; la peculiarità del dispositivo sta nel fatto che esso fornisce *sempre la stessa tensione di uscita*, incurante dei cambiamenti della tensione di ingresso e della corrente di carico.

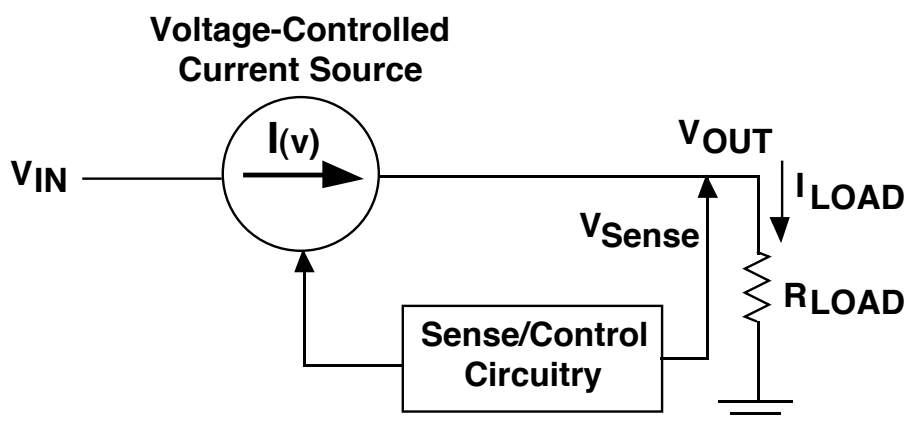


Figura 3.2: Schema funzionale di un regolatore lineare

Come si vede dalla figura 3.2, la rete di controllo ha il compito di monitorare la tensione di uscita al fine di mantenerla costante, regolando la corrente di ingresso; il controllo dell'uscita avviene tramite una rete a retroazione, la quale richiede una compensazione per assicurare la stabilità dell'anello.

Un'ulteriore caratteristica di *ogni* regolatore lineare è che è richiesto un determinato lasso temporale per “correggere” l'uscita dopo un cambiamento della corrente di carico: questo “time lag” si chiama **risposta al transitorio** ed è una misura di quanto velocemente un regolatore è in grado di seguire e compensare eventuali variazioni del carico.

Esistono tre tipologie di convertitori lineari:

- regolatore standard (NPN Darlington);
- regolatore a basso dropout (LDO);

- regolatore quasi-LDO.

Il regolatore lineare di base I primi regolatori di tensione integrati sfruttavano l'uso della configurazione NPN Darlington per il pass-device, ed erano così costituiti come mostrato in figura 3.3.

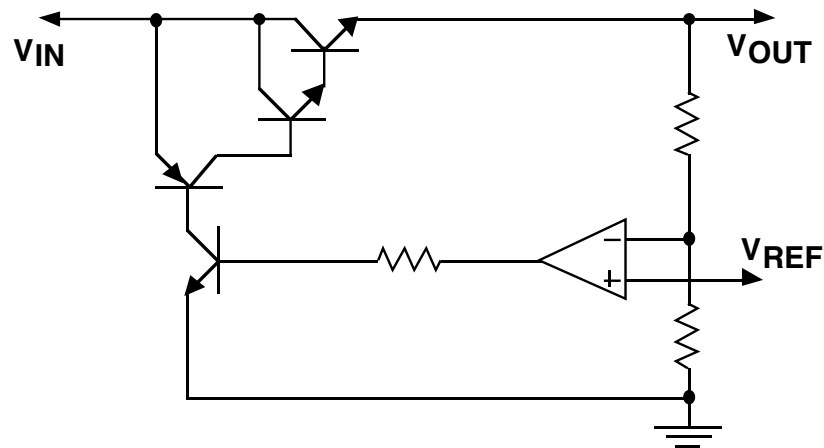


Figura 3.3: Regolatore Standard (NPN)

Un'importante considerazione che va fatta per il regolatore è che, per mantenere la regolazione dell'uscita, il pass-transistor richiede una **tensione minima** su di esso, secondo la relazione:

$$V_{DMIN} = 2V_{BE} + V_{CE} \quad (3.9)$$

Considerando il range di temperature che va da -55°C a $+150^{\circ}\text{C}$, il requisito di tensione minima solitamente di aggira sui 2.5 — 3 Volt.[10] La tensione a cui l'uscita diviene non regolabile (detta *tensione di dropout*) è all'incirca fra 1.5V e 2.2V per un regolatore standard⁵; la tensione di dropout del regolatore standard è la più alta (ovvero la **peggiore**) dei tre tipi presentati; inoltre, la corrente sul pin del riferimento di tensione è molto bassa: ciò avviene perché la corrente di base del pass transistor (la quale scorre sul pin del riferimento) è uguale alla corrente di carico, *divisa per il suo guadagno*. Un pass-device standard⁶ è una rete elettrica composta da un transistor PNP e due NPN: il guadagno complessivo

⁵la tensione di dropout è dipendente sia dalla *corrente di carico*, sia dalla *temperatura*.

⁶con la dicitura *pass device* si intende la rete di comando del pass-transistor.

di corrente risulta quindi molto elevato (> 300); il vantaggio nello sfruttare un pass-device così costituito consiste nel fatto che è richiesta una piccola corrente per polarizzare la base del pass transistor, e ciò risulta in una minore corrente sul pin di riferimento di tensione, che, nel regolatore Standard, è la più bassa delle tre tipologie.

Il regolatore low-dropout (LDO) Il regolatore LDO differisce dalla tipologia Standard poiché il pass device è costituito da un singolo transistor PNP.

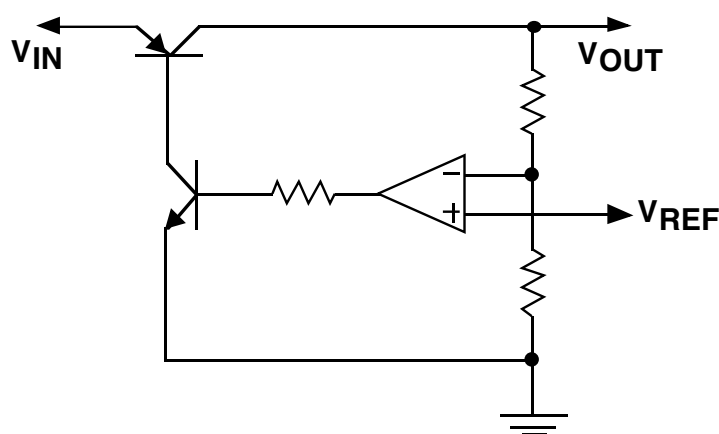


Figura 3.4: Regolatore LDO

La **minima** tensione di drop per mantenere la regolazione dell'uscita consiste solamente della caduta sul transistor PNP:

$$V_{DMIN} = V_{CE} \quad (3.10)$$

La tensione di dropout tipica di un dispositivo siffatto è di 0.7—0.8V a piena corrente di carico (tipicamente è circa 0.6V) ed è direttamente proporzionale alla corrente di carico: per una corrente sufficientemente piccola è possibile ottenere una tensione di dropout di soli 50mV; per questo motivo, il convertitore LDO ha la tensione di dropout più bassa delle tre tipologie (quindi la migliore).

La bassa caduta di questo dispositivo è la principale motivazione della sua diffusione, basti pensare ai dispositivi alimentati a batteria: con questa tipologia di regolatore si ottiene una massimizzazione dell'utilizzo della potenza in ingresso al dispositivo, aumentando quindi il rendimento.

La corrente sul pin del riferimento di tensione coincide in prima approssimazione

alla corrente di carico, divisa per il guadagno del singolo transistor PNP: è la più alta delle tre tipologie.

Il regolatore quasi low-dropout Il regolatore quasi-LDO rappresenta una variante circuitale rispetto al convertitore Standard; il nome quasi-LDO deriva quindi dal fatto che il dropout di tensione è maggiore rispetto ai regolatori LDO, ma inferiore ai regolatori NPN.

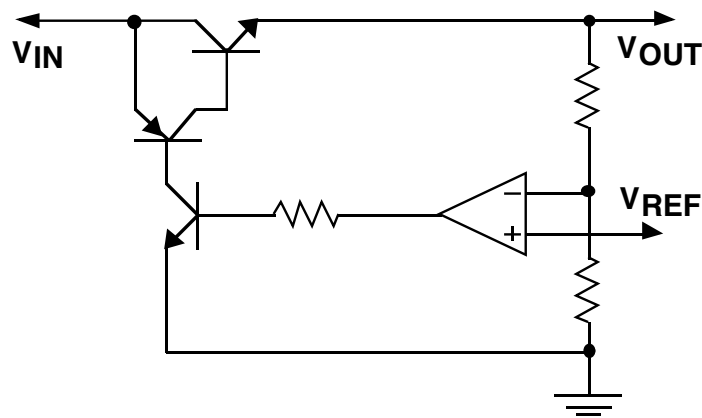


Figura 3.5: Regolatore quasi-LDO

La **minima** tensione di dropout necessaria per mantenere la regolazione dell'uscita è:

$$V_{DMIN} = V_{BE} + V_{CE} \quad (3.11)$$

Tipicamente, la tensione di dropout tipica per questa tipologia di dispositivi si assesta sugli 1.5V massimi; come già specificato precedentemente, la tensione di dropout dipende dalla temperatura e dalla corrente di carico, ma per questi device ci si aspetta di non vederla mai inferiore a 0.9V (a 25°C).

Parametri di scelta del regolatore Il regolatore più adatto per la specifica applicazione può essere scelto considerando i seguenti parametri:

- massima corrente di carico;
- tipologia di tensione in ingresso:

- a Batteria: nelle applicazioni alimentate a batteria, gli LDO sono la scelta *migliore* perché sfruttano in modo più efficiente la potenza in ingresso, potendo quindi operare più a lungo, prima di arrivare allo scaricamento del dispositivo;
- AC: la tensione di dropout non è così critica se la tensione di ingresso DC viene ottenuta per rettificazione di una tensione AC, poichè spesso quest'ultima è regolabile con maggiore facilità; in queste applicazioni, un regolatore Standard è *usualmente* la scelta più economica e può fornire il maggior quantitativo di corrente, anche se le caratteristiche aggiuntive che gli LDO possiedono spesso sono determinanti ed essi divengono la scelta migliore;
- precisione della tensione di uscita: i regolatori lineari hanno una precisione tipica della tensione di uscita nominale del 5% nei casi peggiori; i nuovi regolatori possiedono precisioni inferiori al 2%, a seconda delle condizioni ambientali a cui debbono operare e della corrente di carico
- corrente di quiescenza: la corrente che un dispositivo richiede, anche quando è in stato di *idle*, è di fondamentale importanza nelle applicazioni a batteria: esistono tipi di LDO che richiedono correnti di quiescenza nell'ordine dei 75—150 μA ;
- caratteristiche speciali (pin di shutdown, error flag, così via).

Limitazione in corrente degli LDO La funzione di limitazione in corrente di un LDO ha lo scopo di evitare il danneggiamento del dispositivo se viene connesso un carico con una impedenza nominale troppo bassa: tale impedenza richiederebbe una corrente troppo elevata poiché l'anello di controllo non ha la capacità di bloccarla, portando alla distruzione del pass-transistor; con la funzione di protezione, invece, la rete di limitazione della corrente *prevarica l'anello di controllo*, forzando la corrente di base del pass transistor al massimo valore consentito.

Quando sopraggiunge il limite in corrente (come da figura 3.6), il modo di funzionamento del dispositivo passa da tensione di carico costante a **corrente**

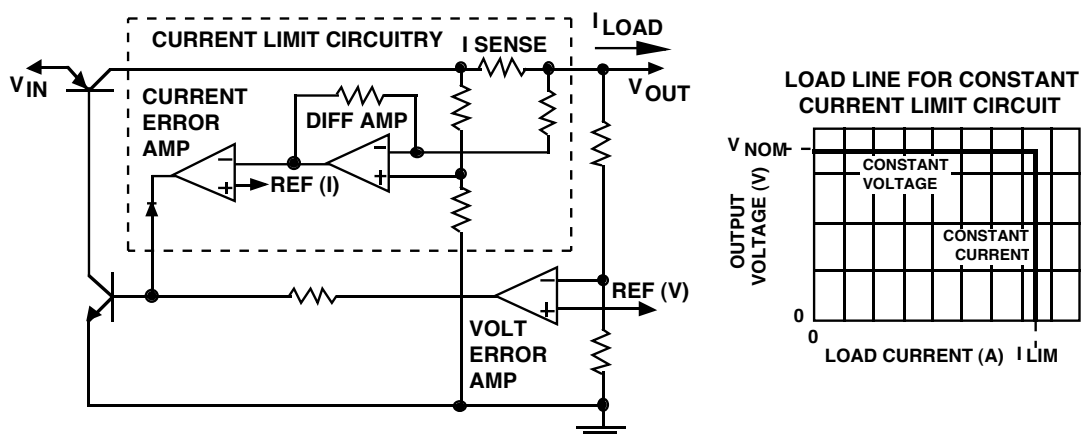


Figura 3.6: Circuito di limitazione a corrente costante

di carico costante: poiché la resistenza di carico cala, essendo sopraggiunto il limite, la caduta della tensione di uscita è direttamente proporzionale alla diminuzione della resistenza di carico.

Influenza del condensatore di uscita sulla stabilità Se vengono commessi errori nella selezione del condensatore di uscita, il circuito può iniziare ad oscillare. Ogni condensatore non ideale possiede degli elementi parassiti indesiderati che ne degradano le performance elettriche; fra questi elementi, i più importanti sono la **resistenza equivalente serie (ESR)** e l'**induttanza equivalente serie**.

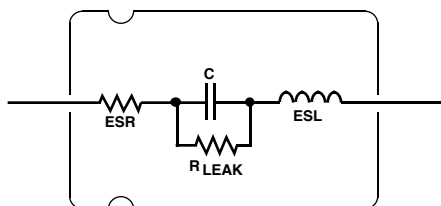


Figura 3.7: Circuito di limitazione a corrente costante

La ESR rappresenta la principale causa di instabilità dell'anello di retroazione sia dei regolatori lineari, sia delle sorgenti SMPS, in quanto introduce una compensazione indesiderata nella funzione di trasferimento uscita-ingresso:

Come si nota dalla figura 3.8, la ESR diminuisce il margine di fase del modulo della funzione di trasferimento uscita-ingresso del sistema introducendo un'azio-

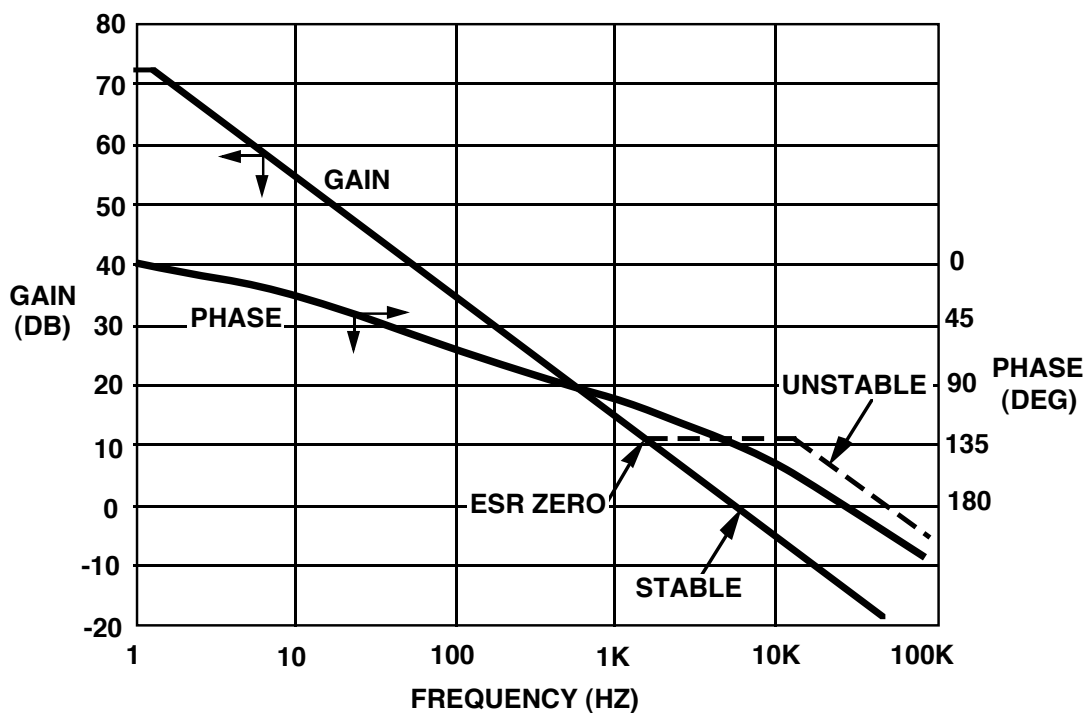


Figura 3.8: Influenza della ESR sulla funzione di trasferimento

ne derivatrice alla pulsazione corrispondete alla frequenza di 2kHz: ciò si traduce nello spostamento della frequenza di attraversamento per l'asse a modulo a guadagno unitario (0 dB) a pulsazioni superiori, determinando instabilità.

La ESL limita l'efficacia del condensatore alle alte frequenze: è per questa motivazione che i condensatori elettrolitici devono essere filtrati da condensatori con migliore risposta RF, quali, ad esempio, i ceramici.

3.2 Dimensionamento dei domini di tensione

Come è stato analizzato nel capitolo due, il sistema è caratterizzato dalla presenza di svariate tensioni di alimentazione indipendenti; per una corretta accensione e un corretto spegnimento del processore AM3892 è necessario che tutte le tensioni vengano fornite in una precisa finestra temporale, nella quale deve essere controllato anche il tempo con cui esse si assestano e la rampa di salita: tale ordine viene denominato *sequenza di power-up* (o *power-down*, nel caso dello spegnimento).

3.2.1 Il powermanager

Dato che i core sono piuttosto sensibili a variazioni delle alimentazioni e che il tempo per sviluppare il prodotto finito è poco, per ottenere una sequenza di power-up accettabile è stato deciso di utilizzare un integrato di gestione delle alimentazioni, o *powermanager*: un dispositivo di questa tipologia non è considerabile un “companion chip” poiché esso è un dispositivo più complesso.

Per la selezione del powermanager ho ricevuto consulenza da Texas Instruments, la quale, dopo un’analisi delle richieste del sistema, ha saputo consigliarmi un prodotto nuovissimo sviluppato appositamente per la classe di processori Sitara ed Integra, il TPS659112[11].

Il dispositivo al suo interno è corredato di tre convertitori abbassatori di tensione, un driver per dispositivi FETs esterni in grado di fornire elevate correnti di carico ed otto LDO: due dei convertitori step-down sono progettati per alimentare il core del processore tramite una tecnologia di scalatura dinamica della tensione (SmartReflex), controllata tramite un’interfaccia I²C⁷; il terzo convertitore è studiato per alimentare gli input/output e la memoria di sistema. In aggiunta alle risorse per l’alimentazione del sistema, il dispositivo contiene un EPC per la gestione della power-up sequence e di un RTC⁸ (la sequenza di accensione è programmabile tramite EEPROM⁹).

⁷Ciò è stato progettato per ottenere un risparmio energetico ottimale.

⁸Sta per Real Time Clock: è un dispositivo con funzione di orologio impiegato all’interno di un computer.

⁹Acronimo di “Electrically Erasable Programmable Read-Only Memory”: è un tipo di memoria non volatile, usata per memorizzare piccole quantità di dati che devono essere mantenuti quando viene tolta l’alimentazione (per esempio la configurazione del dispositivo).

Presento ora una tabella in cui è possibile vedere che potenza eroga ogni alimentazione:

RESOURCE	TYPE	VOLTAGES	POWER
VIO	SMPS	1.5/ 1.8/ 2.5/3.3 V	1500 mA
VDD1	SMPS	0.6 ... 1.5 V in 12.5-mV steps Programmable multiplication factor: x2, x3	1500 mA
VDD2	SMPS	0.6 ... 1.5 V in 12.5-mV steps Programmable multiplication factor: x2, x3	1500 mA
VDDCtrl	SMPS	0.6 ... 1.4 V in 12.5-mV steps	6000 mA
LDO1	LDO	1.0–3.3 V, 0.05-V step	320 mA
LDO2	LDO	1.0–3.3 V, 0.05-V step	320 mA
LDO3	LDO	1.0–3.3 V, 0.1-V step	200 mA
LDO4	LDO	1.0–3.3 V, 0.05-V step	50 mA
LDO5	LDO	1.0–3.3 V, 0.1-V step	300 mA
LDO6	LDO	1.0–3.3 V, 0.1-V step	300 mA
LDO7	LDO	1.0–3.3 V, 0.1-V step	300 mA
LDO8	LDO	1.0–3.3 V, 0.1-V step	300 mA

Figura 3.9: Sorgenti di alimentazione e correnti erogabili

La evaluation board del processore AM3892 fornitami si è distinta per la sua completezza di dotazione anche perché è stata l'unica delle tre EVB a possedere un *power monitor*: tale microcontrollore (un MSP430 di Texas, un microcontrollore a 16 bit, ultra low power) si occupa di monitorare tutte le tensioni e le correnti in uscita dai convertitori e direzionare l'output ad una porta seriale.

Come si vede dalla figura 3.10, con un accesso tramite Putty (è un client SSH, Telnet ed rlogin combinato con un emulatore di terminale: ha la stessa funzione di Minicom per *nix), l'MSP430 invia alla seriale i consumi istantanei, che sono piuttosto stabili per le cifre più significative, quindi attendibili ai fini del calcolo del consumo di potenza del sistema completo (sistema Android avviato più applicativo del cliente "up & running").

La figura 3.11 illustra l'ordine di avvio dei convertitori e gli slot temporali in cui avviene il settling delle tensioni.

3.2.2 Generazione della tensione 5V

Il dispositivo di gestione delle alimentazioni si fa carico di generare quasi tutte le tensioni operative partendo dai 5V mentre la board, da specifica, accetta in ingresso 12V: è quindi necessario abbassare tale tensione, mantenendo una corrente di carico elevata.

3.2 DIMENSIONAMENTO DEI DOMINI DI TENSIONE

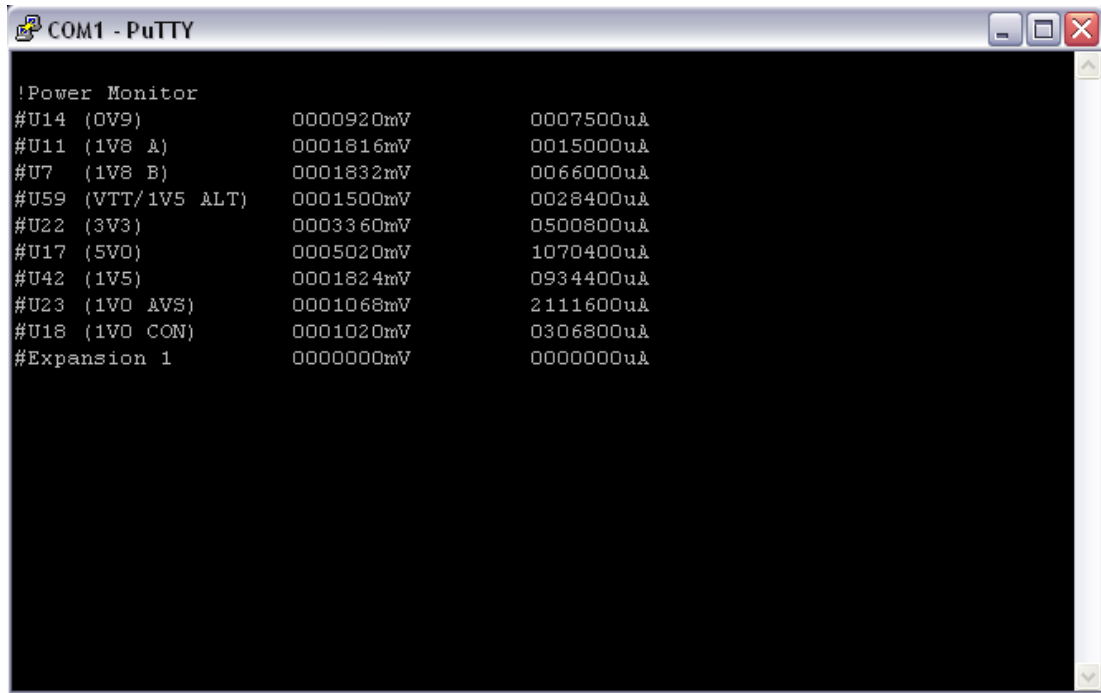


Figura 3.10: Consumi del sistema osservati tramite porta seriale

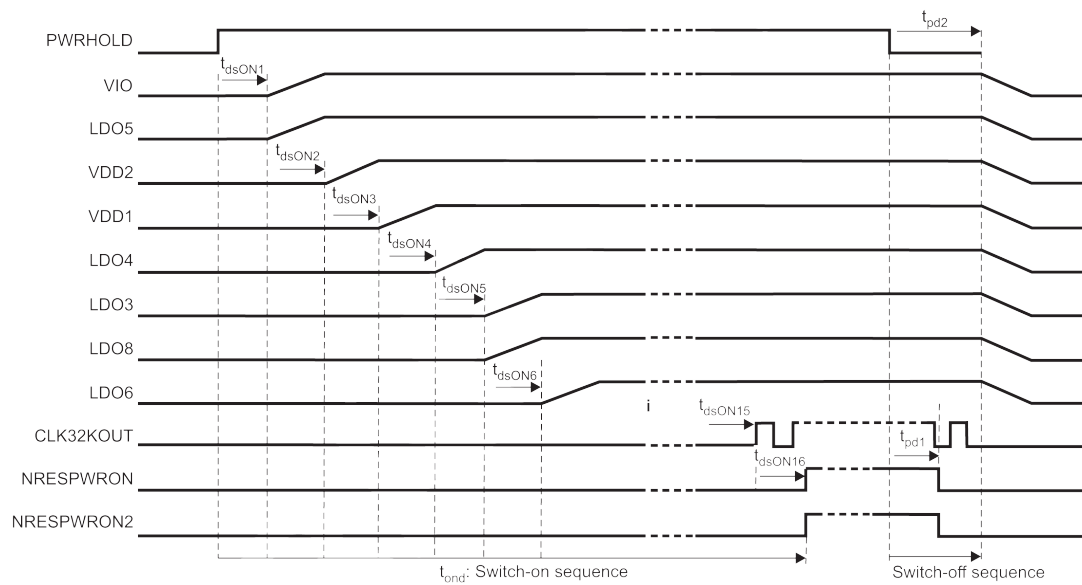


Figura 3.11: Sequenza di power-up del sistema del powermanager TPS659112

A questo scopo è stato scelto il convertitore step-down TPS54620 di Texas Instruments.

3 DIMENSIONAMENTO DELLO STADIO DI ALIMENTAZIONE

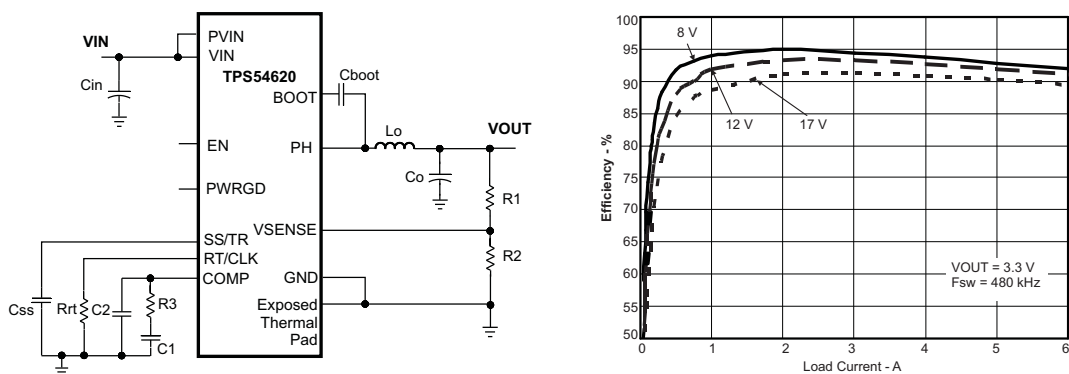


Figura 3.12: Schema semplificato e retta di carico del TPS54620

Come si vede dal grafico in figura 3.12, osservando il punto di carico in cui gravita il sistema progettato, il convertitore ha un'efficienza del 94%; il partitore resistivo in uscita, responsabile della configurazione della tensione di carico, è stato dimensionato per ottenere un valore di tensione lievemente superiore (5.08V); per aiutarmi nel dimensionamento dell'alimentatore, ho realizzato un foglio di calcolo contenente tutte le formule di interesse (tensioni di ingresso, tensioni di uscita, correnti di carico, valori di induttanza, capacità di ingresso, capacità di uscita, calcolo ESR, calcolo della compensazione in frequenza) e lo riporto, per completezza, alla prossima pagina.

3.2 DIMENSIONAMENTO DEI DOMINI DI TENSIONE

Dati iniziali	Dimensionamento parametri del convertitore	Compensazione in frequenza	
V _{in_min} = 11,80 V	VSTART = 6,528 V	C _{out} = 68,00 uF	0,000068000000 F
V _{in_max} = 12,20 V	VENFALLING = 1,170 V	f _{p_mod} = 2,76 kHz	2758,955444791900 Hz
V _{out} = 5,09 V	VENRISING = 1,210 V	f _{z_mod} = 9,20 kHz	9196,52 Hz
deltaV _{out} = 5,00 %	VSTOP = 6,190 V		
V _{ref} = 0,80 V	lp = 1,15 uA	f _{co_1} = 5,04 kHz	5037,140441 Hz
I _{out} = 6,00 A	lh = 3,40 uA	f _{co_2} = 25,73 kHz	25732,26 Hz
	Kind = 0,30		
	R _{rt} = 99,87 kOhm	f _{co} = 5,04 kHz	5037,140441 Hz
F _{sw} = 480,00 kHz			
C _{ss} = 10,00 nF	T _{ss} = 3,48 ms	R ₂ = 877,76 Ohm	877,76 Ohm
I _{ss} = 2,30 uA		C ₃ = 65720,28 pF	0,000000065720 F
C _{in} = 10,00 uF	L ₁ = 3,43 uH	g _{m_ea} = 1300 uA/V	0,001300000000 A/V
	I _{ripple} = 1,80 A	g _{m_ps} = 12,00 A/V	
	I _{L_rms} = 6,02 A		
	I _{L_peak} = 6,90 A		
	C _{o_spec1} > 29,47 uF	C _p = 18,00 pF	0,000000000018 F
	C _{o_spec2} > 1,84 uF	f _p = 0,00 kHz	0,000010073296 Hz
	Co > 29,47 uF		
	R _{esr} < 254,50 mOhm		
	I _{co_rms} = 519,62 mA		
	I _{ci_rms} = 2,97 A		
	delta_V _{in} = 281,25 mV		
	C _{slowstart} = 10,00 nF		
	R ₅ = 53,63 kOhm		53625,000000000000 Ohm
	R ₁ = 35,54 kOhm		35543,27 Ohm
	R ₂ = 8,03 kOhm		8025,44521 Ohm

3 DIMENSIONAMENTO DELLO STADIO DI ALIMENTAZIONE

Capitolo 4

Schematici di progetto

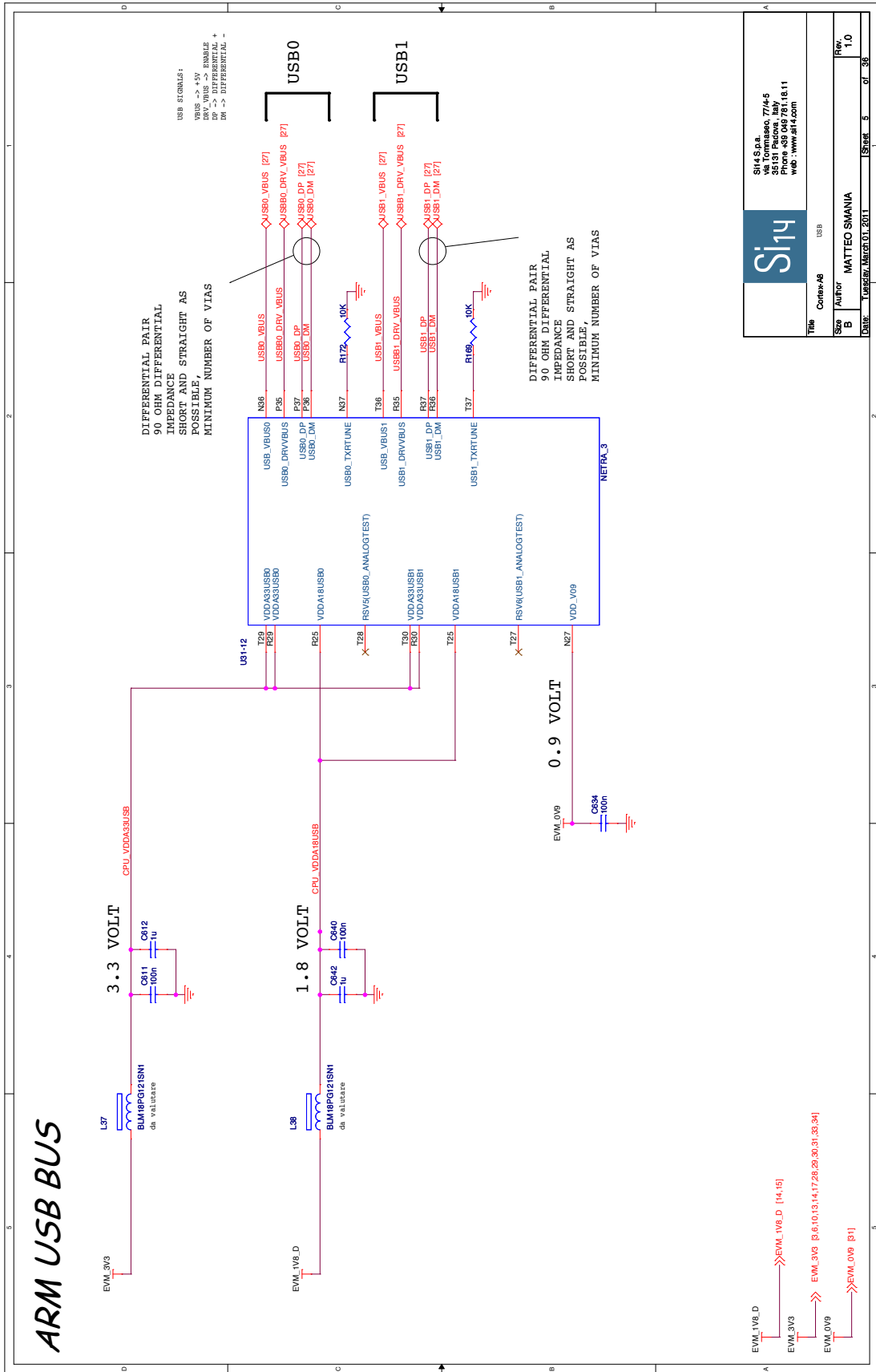
Il questo capitolo sono raccolti alcuni schematici di interesse del progetto, per far meglio comprendere come effettivamente è stato realizzato il sistema rispetto alla descrizione che è stata data nel capitolo 2.

4.1 Schematici

Gli schematici sono proposti pagina per pagina, qui a seguire; non è presente la totalità di essi perché alcuni sono o di scarso interesse (riferimenti a massa, test points, fori meccanici e di allineamento) o ridondanti.

Gli schematici sono utili anche per mostrare come è stato dimensionato lo stadio di alimentazione e che convertitori sono stati impiegati.

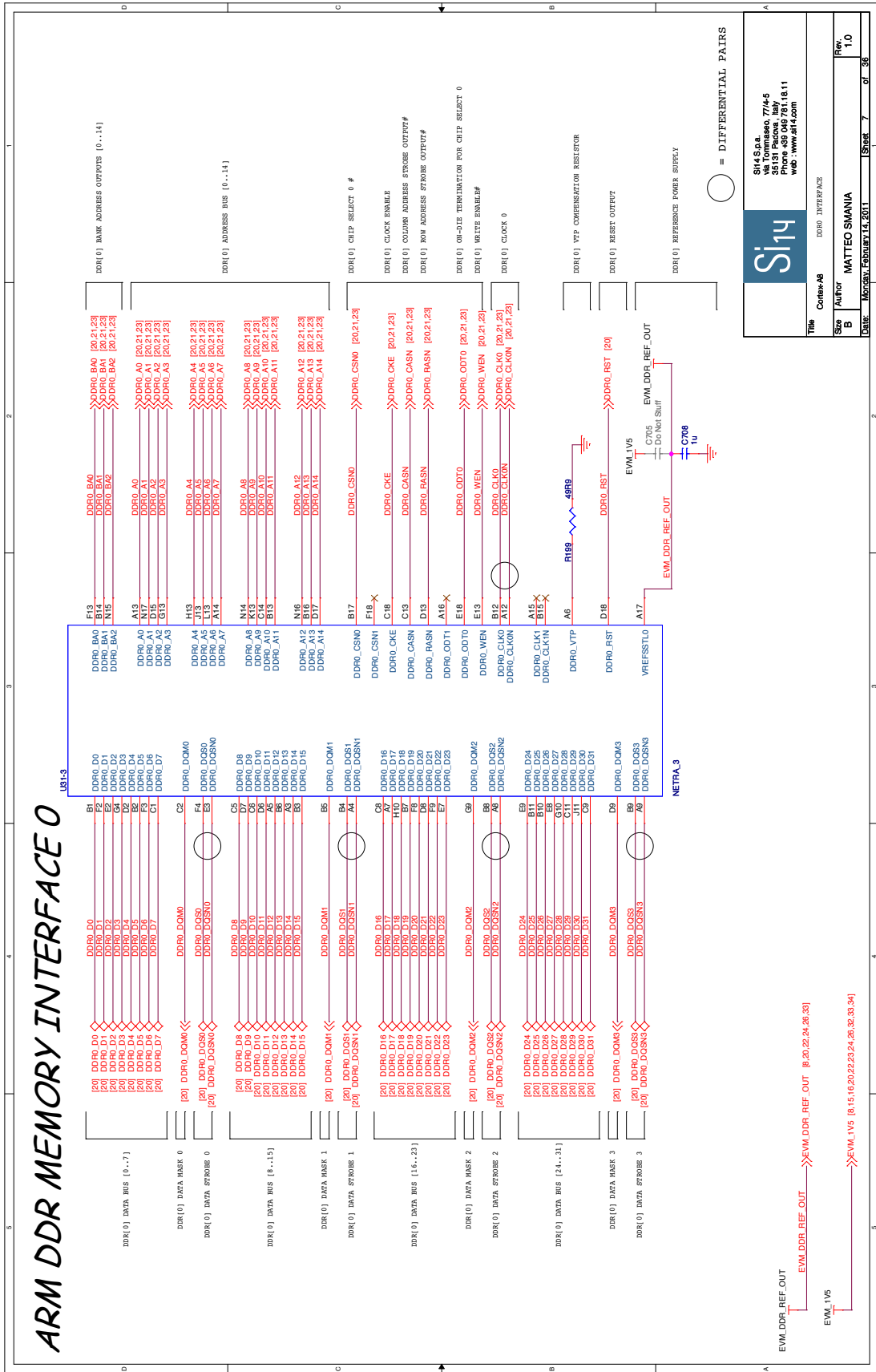
4.1 SCHEMATICI



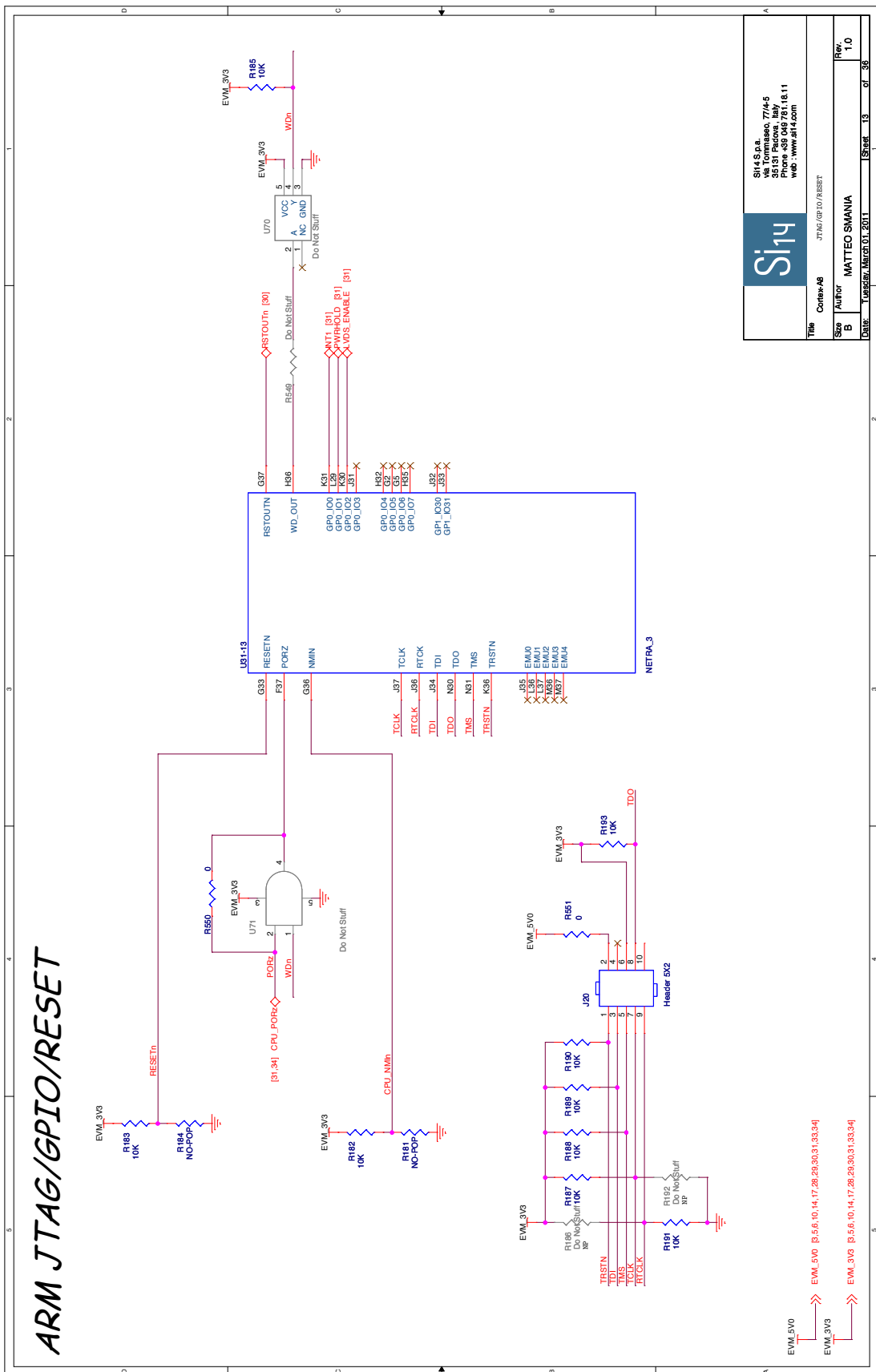
SH4 S.p.A.
 Via Tommaso 7/A-5
 37060 S. Pietro Lupatoto (VI)
 Phone: +39 0445 7811811
 web: www.sh4.com

SH4

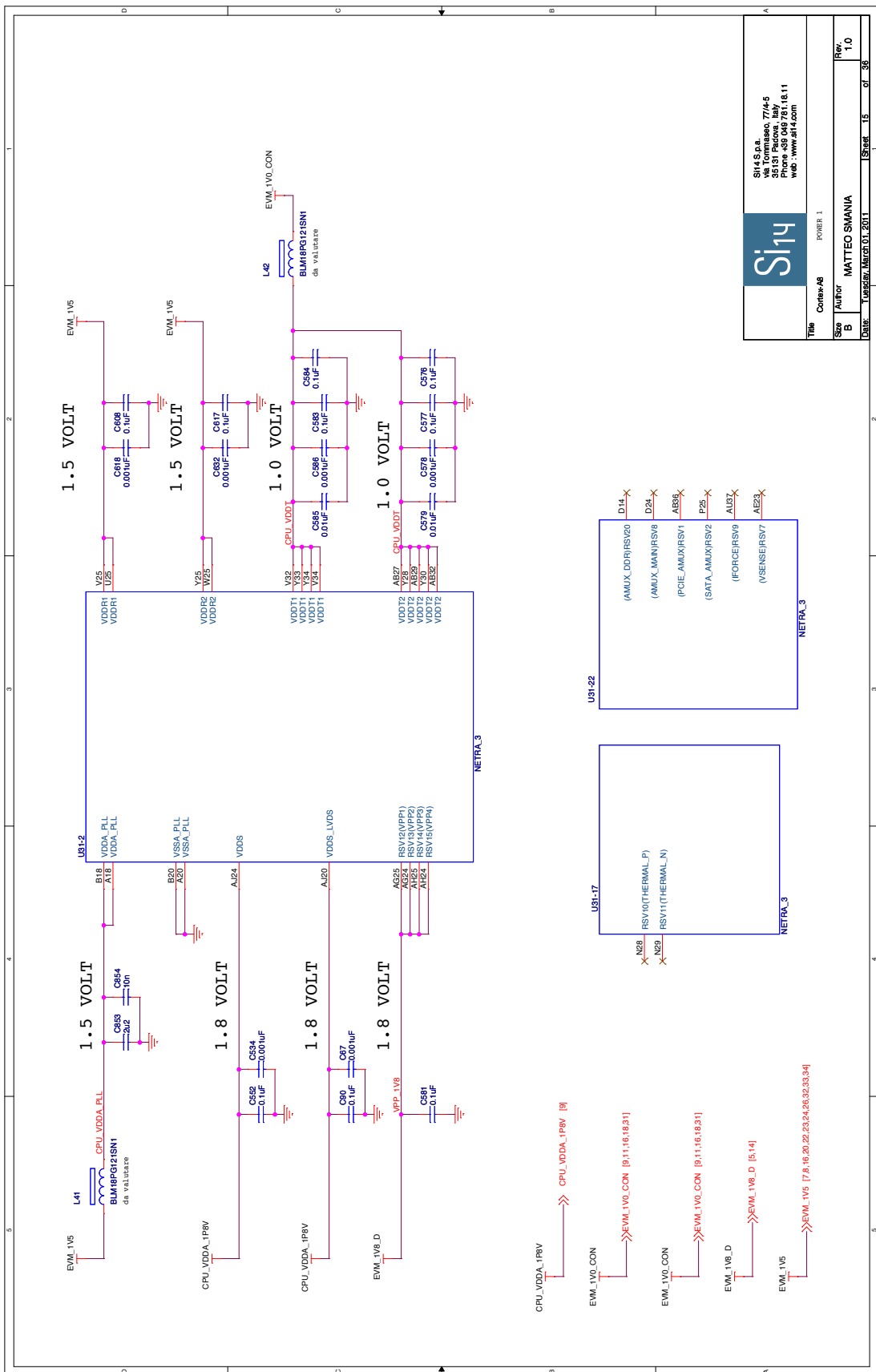
Title: Cortex-A8 USB
 Size: B
 Author: MATTEO SMANIA
 Date: 10/09/2011
 Sheet: 5 of 36
 Rev: 1.0



4 SCHEMATICI DI PROGETTO



4 SCHEMATICI DI PROGETTO

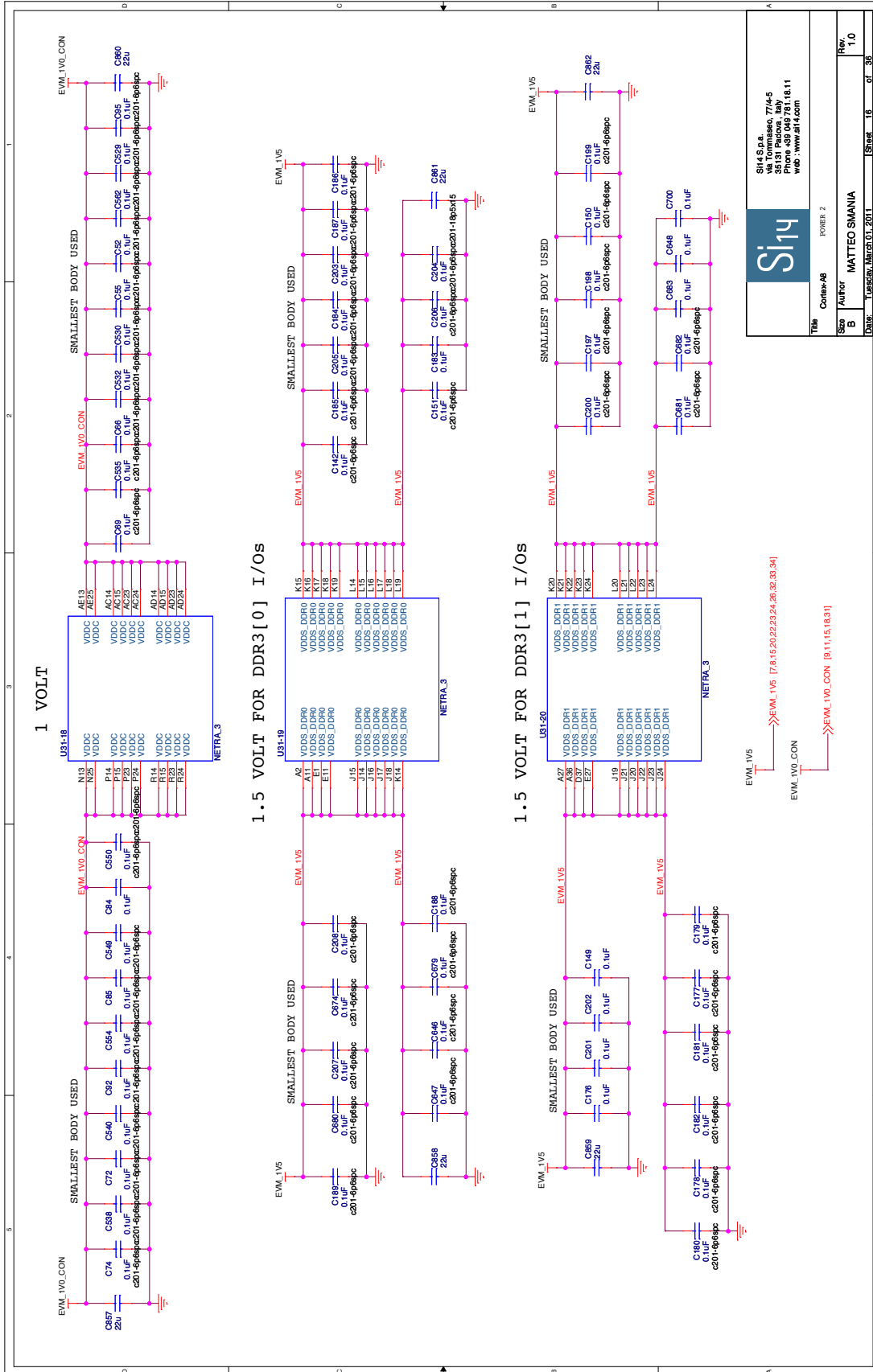


SH14

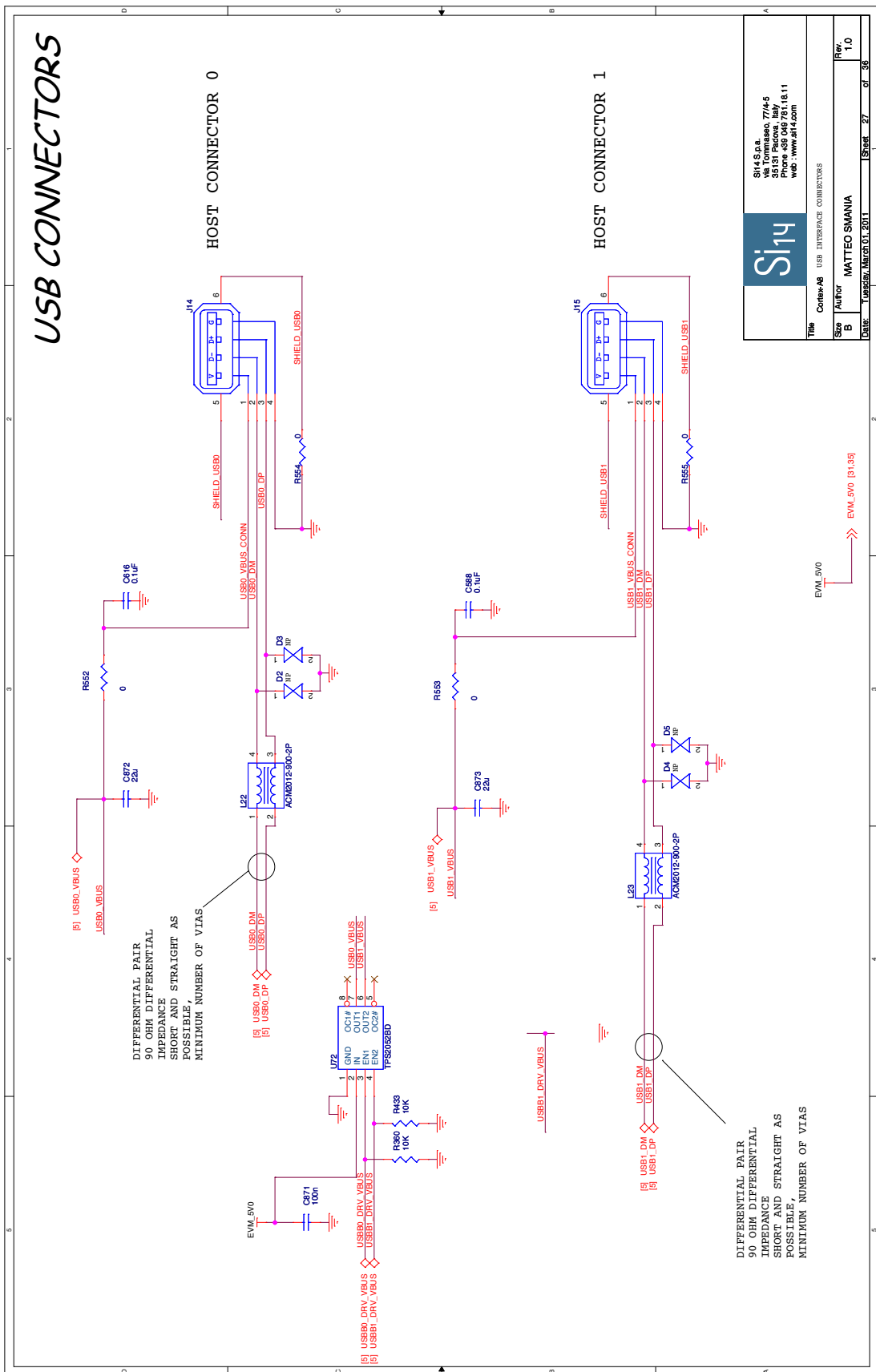
SH14 S.p.a.
 Via Tommaso 7/A-5
 00198 Roma (RM)
 Phone: +39 06 781 18.11
 web: www.sh14.com

Titolo: Cortex-A8 POWER 1
 Autore: MATTEO SMANIA
 Data: 15/05/2011
 Foglio: 15 di 36

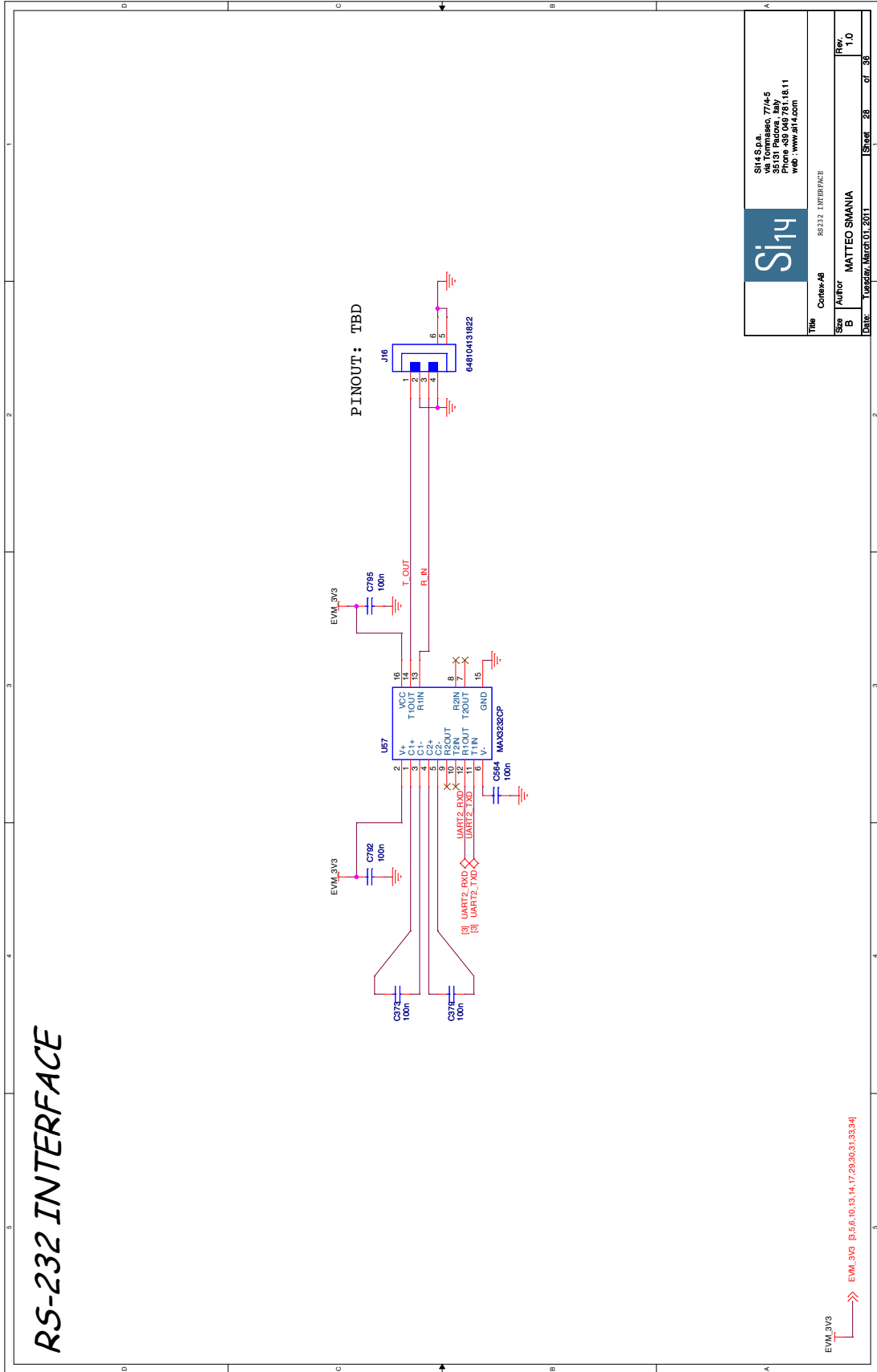
4.1 SCHEMATICI



4 SCHEMATICI DI PROGETTO



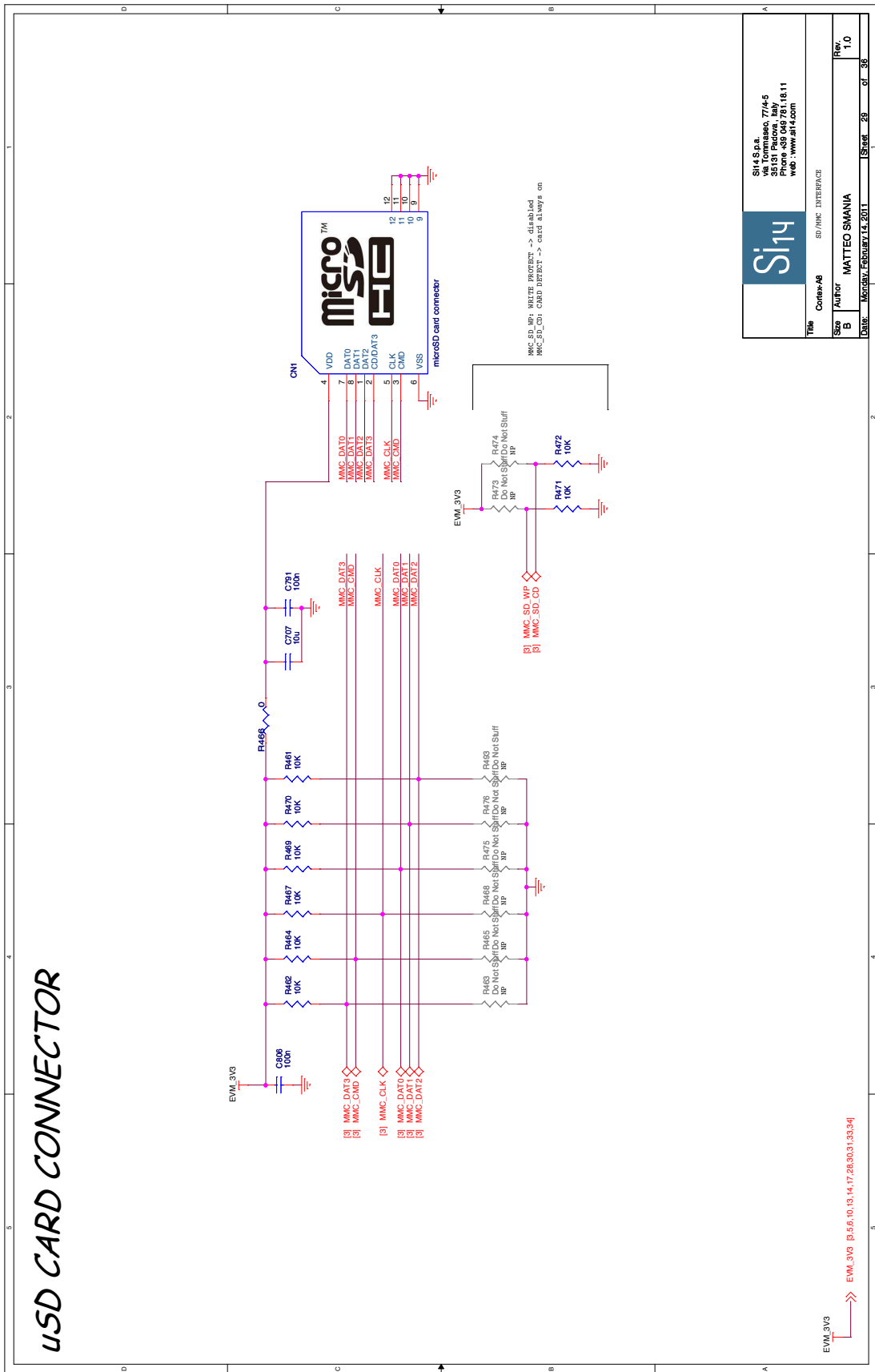
Si14	
SH14 S.p.A. Via Tommaso, 7/A-5 00144 Roma, Italia Phone: +39 06 781 18.11 Fax: +39 06 781 18.11 Web: www.si14.com	
USB INTERFACE CONNECTORS	
Titolo	Context-A8
Scale	B
Autore	MATTEO SMANIA
DATA:	TURIN, 27 MARZO 2011
Stampa	27
Rev.	1.0
di	36



Si14		Si14 S.p.a. Via Tommaso 7/A-5 00144 Roma Phone: +39 06 781 18.11 Web: www.si14.com	
Title	Context: A8	RS232 INTERFACE	
Size	B	Author	MATTEO SMANIA
Date	TUESDAY, MARCH 01, 2011	Sheet	28 of 36
		Rev	1.0

EVM_3V3 → EVM_3V3 [5,5,6,10,13,14,17,28,30,31,33,34]

4 SCHEMATICI DI PROGETTO

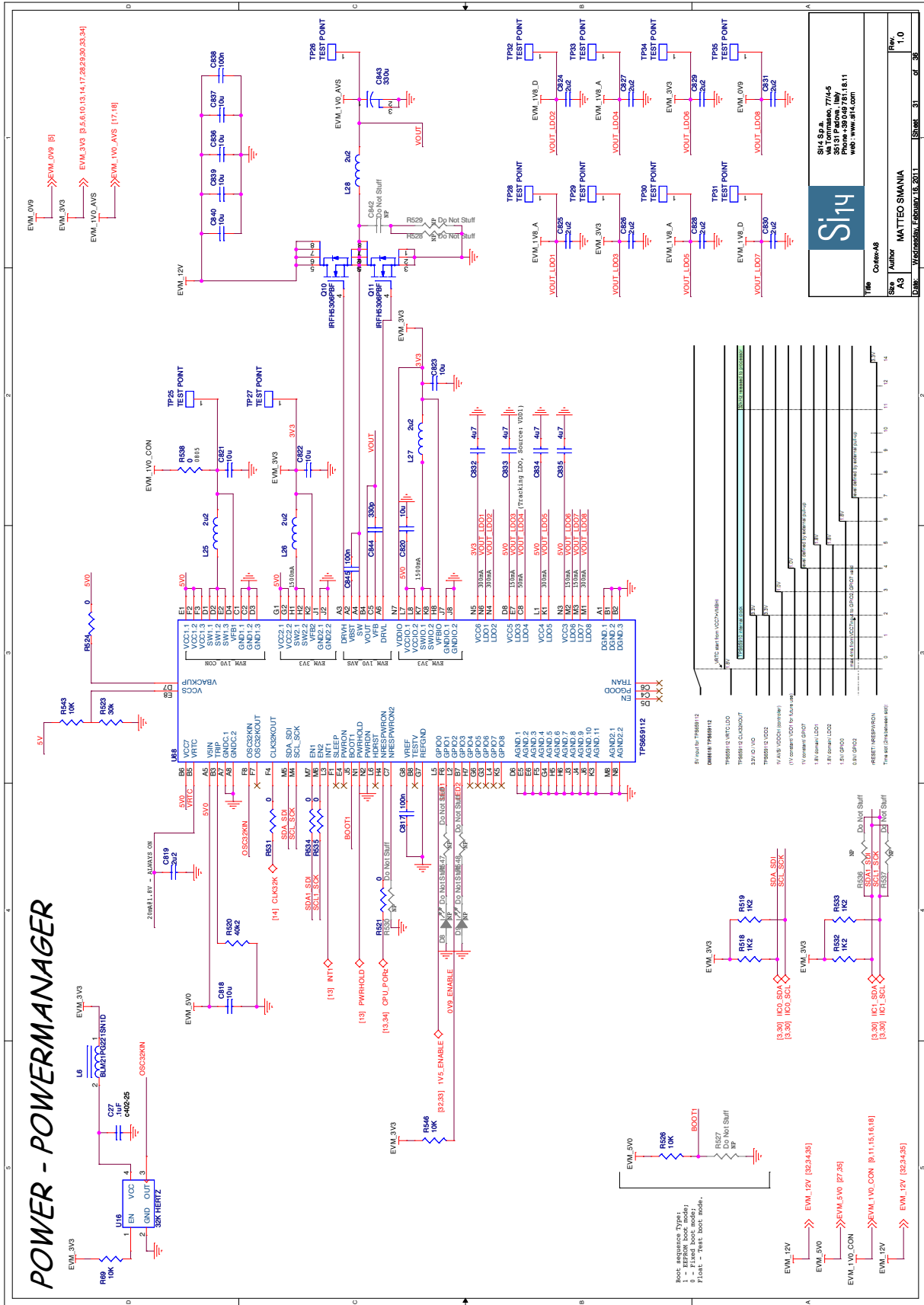


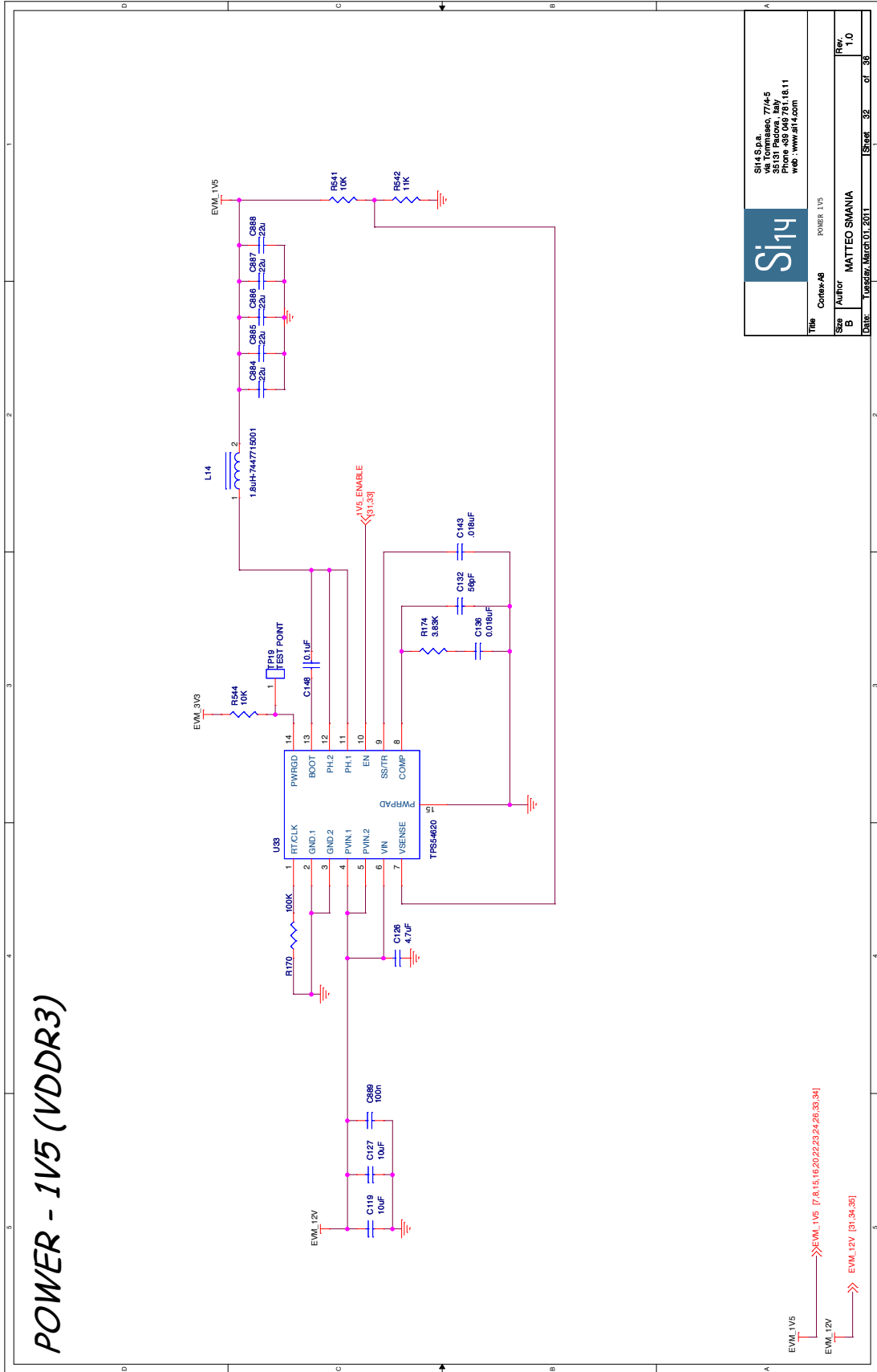
uSD CARD CONNECTOR

EVM_3V3

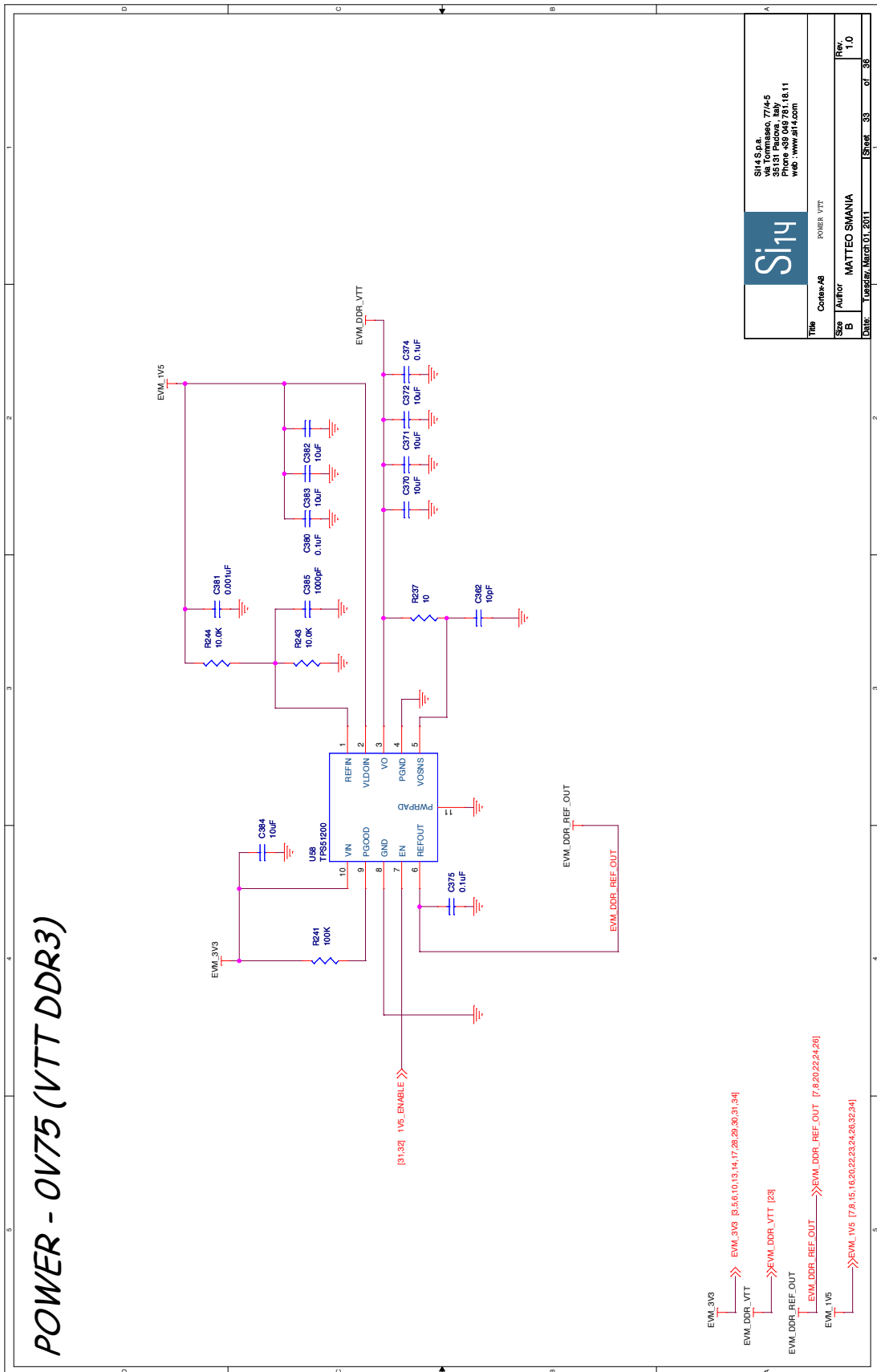
EVM_3V3 [5,6,10,13,14,17,26,30,31,33,34]

4 SCHEMATICI DI PROGETTO

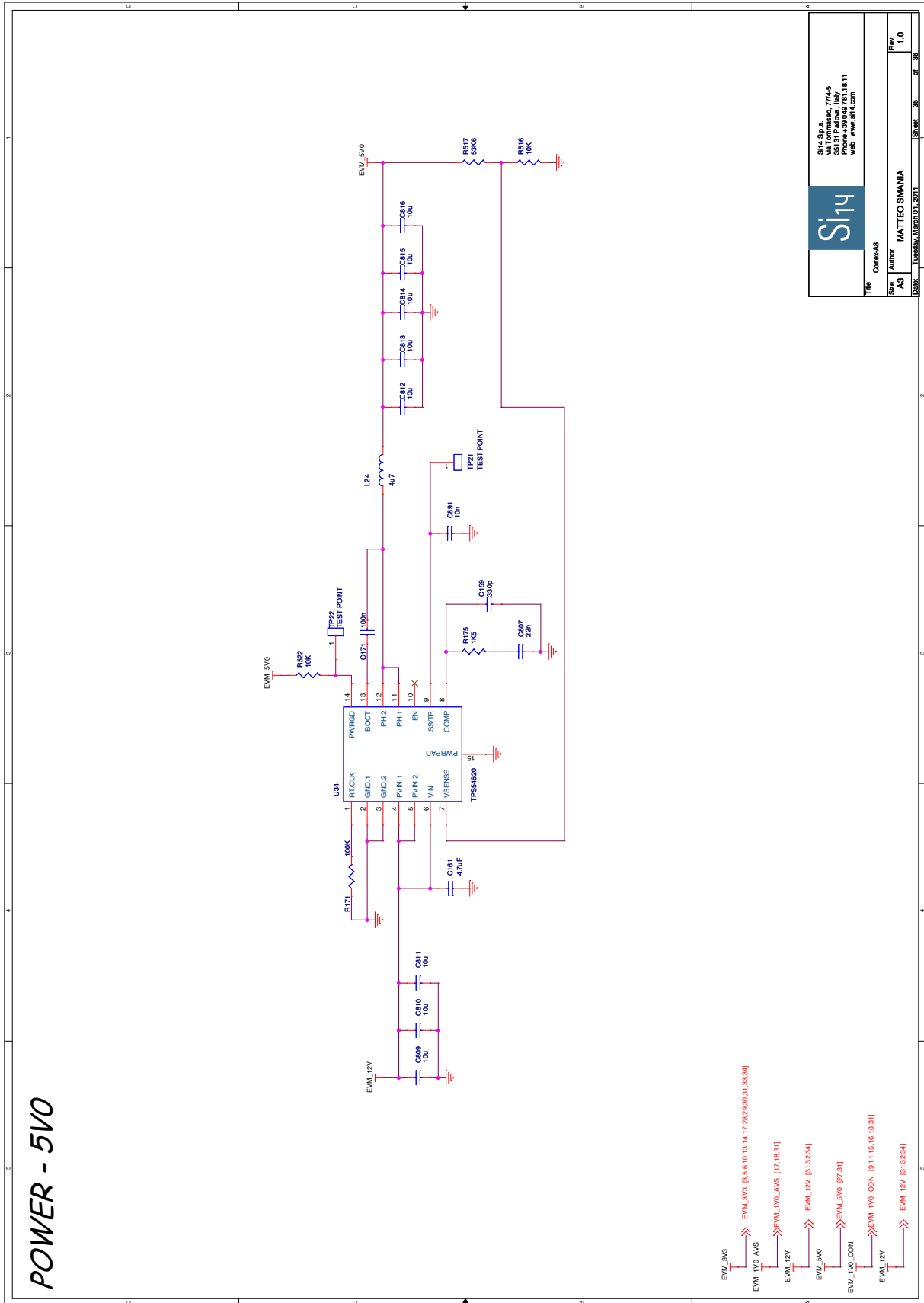




4 SCHEMATICI DI PROGETTO

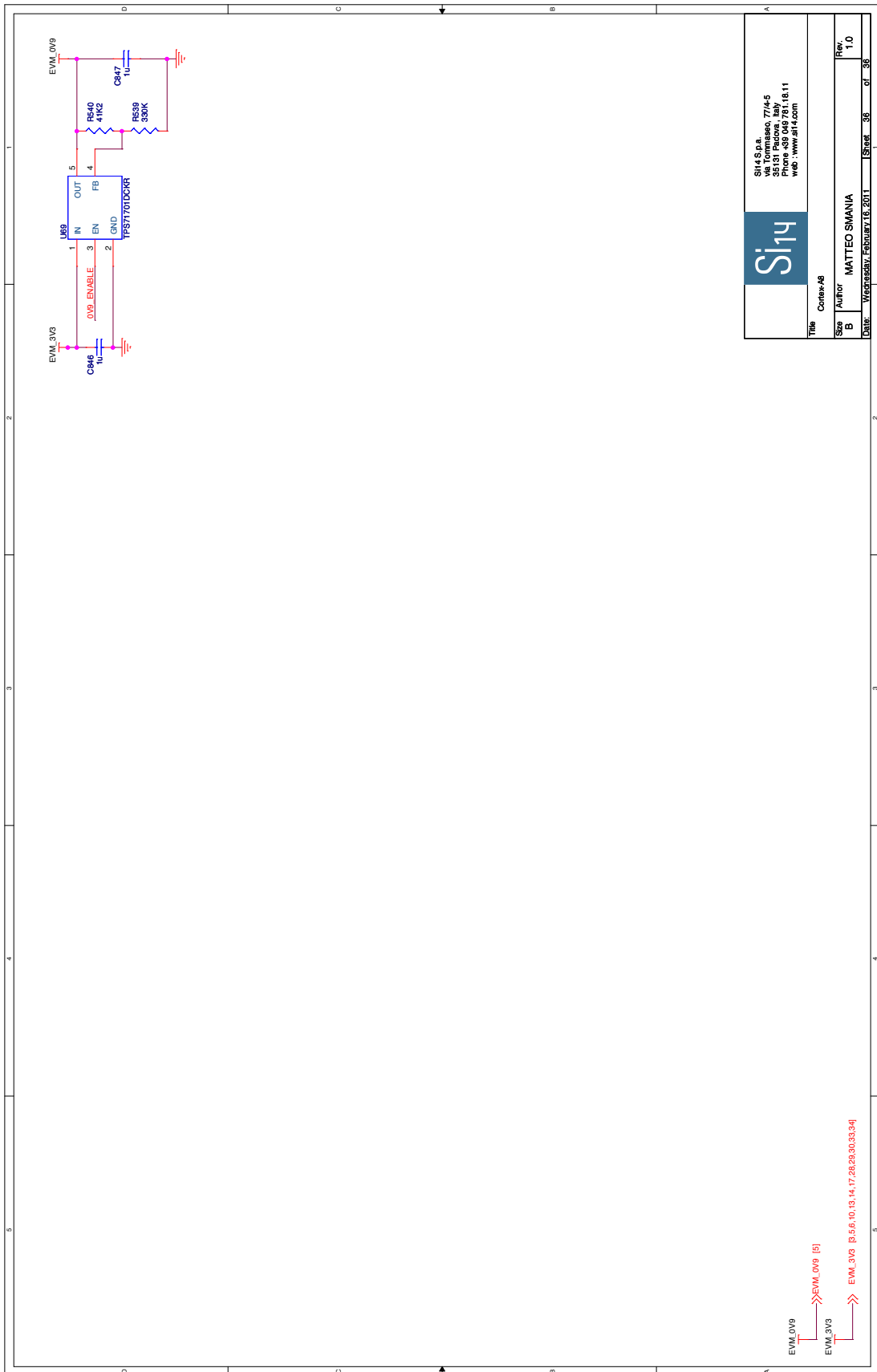


Si14		SI14 S.p.a. Via Tommaso 7/A-5 00100 Roma Phone: +39 06 781 18.11 Web: www.si14.com	
Title	POWER_VTT	Sheet	35 of 36
Size	B	Rev	1.0
Author	MATTEO SMANIA		
Date	TUESDAY, MARCH 01, 2011		



		SIM S.p.A. Via Tommaso, 774-5 35031 Padova, Italy Tel: +39 049 871111 Web: www.si14.com	
		Tipo: Corrente-A8	Data: 10/05/2011
Size: A3	Author: MATTEO SIMANIA	Rev: 1.0	Cl: 35
DWG: 10/05/2011	Sheet: 35	of: 35	

4 SCHEMATICI DI PROGETTO



Capitolo 5

Considerazioni sul layout e sulle emissioni elettromagnetiche

In questo capitolo verranno illustrati alcuni concetti fondamentali sulla natura dei segnali che vengono impiegati nei circuiti integrati di tipo analogico e digitale, nonché degli accorgimenti che sono indice di una *buona progettazione*.

Il progettista non termina il suo operato dopo aver realizzato gli schematici di progetto e convertito lo schematico in un layout funzionale: il passo successivo è accertarsi che ciò che è stato realizzato sia in grado di poter coesistere con altre apparecchiature elettroniche, nell'ambiente di destinazione; di fatto, la board costruita *non deve disturbare elettromagneticamente le altre apparecchiature e deve anche essere robusta rispetto ai sicuri disturbi che le altre apparecchiature generano*.

Apparentemente questa potrebbe sembrare una sezione del progetto *banale*, in realtà i concetti qui elencati sono proprio quelli che consentono di produrre circuiti elettrici conformi alle normative, eliminando (o perlomeno riducendo) il ricorso ad accorgimenti aggiuntivi una volta terminato il circuito (come, per esempio, l'impiego di *schermi* o altri dispositivi, spesso costosi e di difficile introduzione nel prodotto finito).

5.1 Tecniche di layout conformi alle normative

5.1.1 Il riferimento dei potenziali

Definizione di massa

Si definisce come *massa* quella parte circuitale che serve come riferimento dei potenziali in un generico circuito elettronico; dal punto di vista della rappresentazione grafica, essa viene indicata negli schemi elettrici con vari simboli; normalmente si considera che in un generico circuito i vari punti “a massa” siano equipotenziali e che attraverso di essa possa circolare qualsiasi corrente: comprendiamo subito che questa definizione è di uso comune, ma non si presta ad un’utilizzo ingegneristico.

Introduciamo quindi una definizione più consona: la massa è una parte circuitale *a bassa impedenza* predisposta per consentire ad una corrente di ritornare alla sorgente dalla quale è partita. La definizione presentata va ulteriormente precisata: per *bassa impedenza* del circuito di massa si intende bassa rispetto alle altre parti del circuito interessate. Uno degli errori più comuni in fase di progettazione è il non tener conto che una qualsiasi corrente elettrica, per poter circolare, richiede un circuito chiuso: le correnti in un circuito elettronico, dopo aver espletato la loro funzione nelle varie parti del circuito, vanno “a massa”, senza la possibilità di individuare con dettaglio il percorso che esse scelgono per tornare al generatore. Un ulteriore aspetto importante è che la *massa* non deve essere confusa con la *terra*: col termine di *terra* si intende un elemento circuitale che possiede un potenziale di 0V; in un circuito elettronico la massa è sempre presente, mentre può non essere necessaria la terra.

Risulta piuttosto complesso modellare il comportamento reale di un circuito con attenzione al percorso di massa: sono spesso trascurati i componenti parassiti, i quali però sono **sempre** presenti: è importante, quindi, comprendere in che condizioni è possibile accettare come equipotenziali due punti su un percorso conduttivo attraversato da corrente (e dunque, non equipotenziale).

Riferimento dei potenziali per i sistemi digitali

Nella letteratura sono riportati molti accorgimenti volti a costruire buoni circuiti digitali, però questi hanno una natura limitata, in quanto il parametro fonda-

mentale che coinvolge i disturbi elettromagnetici da tenere sotto osservazione è *la frequenza di funzionamento del circuito*, che è un esempio di grandezza elettrica per cui la velocità di variazione è notevole: gli accorgimenti da adottare, quindi, dipendono dal contesto operativo.

Si potrebbe pensare, ad esempio, che un accorgimento utile al fine di evitare che un segnale venga sporcato da rumore sia quello di ricorrere ad un *filtraggio* mediante un condensatore¹: si noti che un terminale del condensatore non è connesso al riferimento ed è condiviso con altre parti del circuito, quindi, se supponiamo che esista una corrente ad alta frequenza proveniente da altri circuiti, la sua presenza favorisce l'immissione delle componenti HF nel circuito che si voleva proteggere. Il problema di connettività più difficile da risolvere è sempre quello di ottenere una "buona massa": l'oscilloscopio mostra rumore se viene a mancare una buona massa, ma è difficile garantire una buona connessione al riferimento poiché i pin del circuito integrato sono sottilissimi (e spesso veicolano segnali ad alta frequenza[2]).

¹supponendo che tale rumore sia in una banda di frequenza diversa dal segnale utile.

5.1.2 Influenza della non idealità della massa

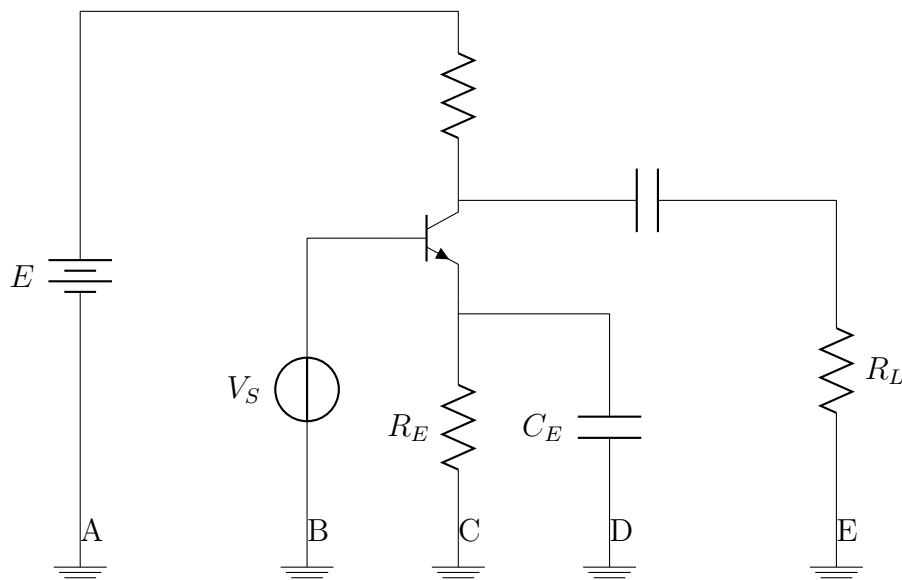


Figura 5.1: Schema di principio di un amplificatore

La prima distinzione che va fatta per mettere in luce le nonidealità del percorso di massa è fra la componente continua della corrente e le componenti alternate: la distinzione risulta necessaria in quanto l'interesse maggiore nello scostamento dall'idealità è dovuto alle componenti variabili.

Il carico può essere percorso solo da componenti variabili per la presenza della capacità in serie, mentre la resistenza di emettitore è, in prima approssimazione, percorsa solo dalla componente continua per via della capacità di emettitore, disposta in parallelo, che cortocircuita le componenti variabili: da questo piccolo esempio si può osservare che a massa giungono componenti fra loro diverse, ma che tutte devono ritornare al generatore; considerando il percorso di ritorno CBA, notiamo che il punto C, per richiudersi al punto A, passa per B, punto in cui passa la corrente del generatore di segnale; a caduta di tensione costante V_{CB} è completamente sentita dal circuito di ingresso e ciò comporta uno *spostamento del punto di lavoro* del transistor, il quale può avere svariate conseguenze, quale una variazione dell'amplificazione complessiva, ad esempio.

Anche la corrente alternata proveniente dal carico circola sul tratto BC di massa: in questo caso, la caduta di tensione è in parte interpretata come parte del segnale inviato all'ingresso; le conseguenze che questo feedback può provocare sono un'o-

scillazione dell'amplificatore (rendendo, di fatto, inutilizzabile il circuito!), oppure un deterioramento del SNR², portando ad un peggioramento delle prestazioni del circuito.

La struttura multipoint della massa

Un metodo per cercare di attenuare le problematiche descritte nel precedente paragrafo è quella di utilizzare un unico elemento conduttore per tutti i componenti del circuito con dimensioni tali da presentare un'impedenza trascurabile per il sistema in cui è inserito.

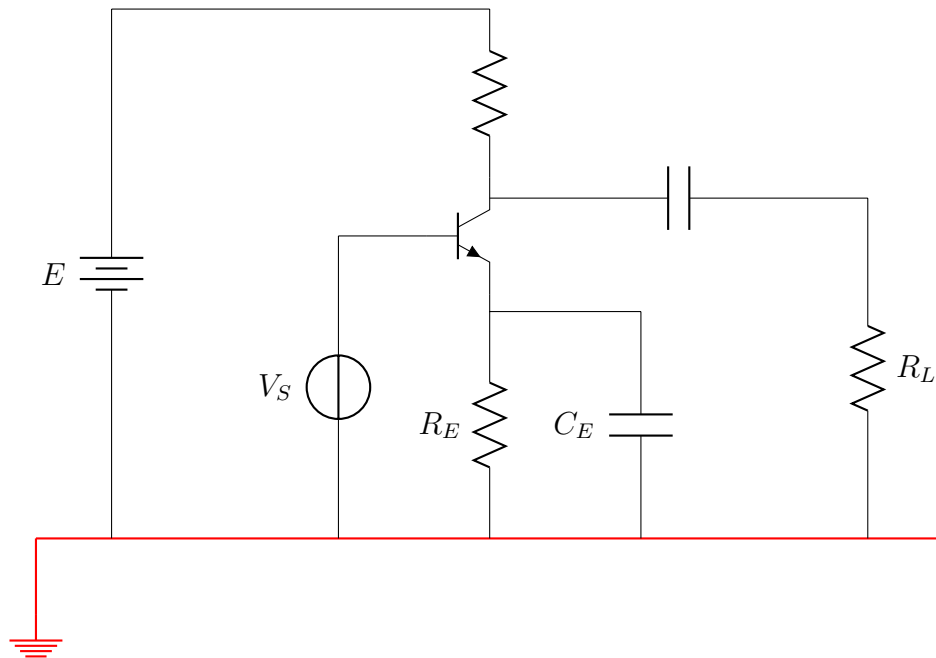


Figura 5.2: Schema di massa di tipo multipoint

In questa configurazione ha molta importanza la frequenza dei segnali che percorrono il circuito:

- in *bassa frequenza*, fino al kHz, ha interesse la resistenza del conduttore di massa, quindi aumentandone la sezione, essa diminuisce;
- in *alta frequenza*, gradualmente si fa sentire l'*effetto pelle* il quale comporta un incremento della resistenza; inoltre gradualmente si fa sentire il compor-

²SNR sta per "Signal to Noise Ratio", ovvero il rapporto fra il segnale utile ed il rumore.

5 CONSIDERAZIONI SUL LAYOUT E SULLE EMISSIONI ELETTROMAGNETICHE

tamento induttivo del conduttore di massa, rendendo necessario prestare attenzione non più alla sezione del conduttore di massa, quanto la *geometria* del sistema.

Lasciando inalterato il circuito dell'esempio 5.1 dato che è un amplificatore, si suppone che le correnti in uscita siano maggiori di quelle in ingresso, è possibile quindi modificare il percorso di massa affinché tali correnti non debbano ritornare all'ingresso.

Riassumendo ciò che è stato analizzato, per organizzare un percorso di massa si può così procedere:

- è necessario individuare i percorsi di ritorno delle correnti nel circuito in esame;
- va valutata l'influenza delle cadute di tensione provocate dalle correnti nei vari tratti del circuito di massa;
- si devono adottare gli opportuni accorgimenti che permettano di annullare (o almeno diminuire) le cadute di tensione sul circuito di massa.

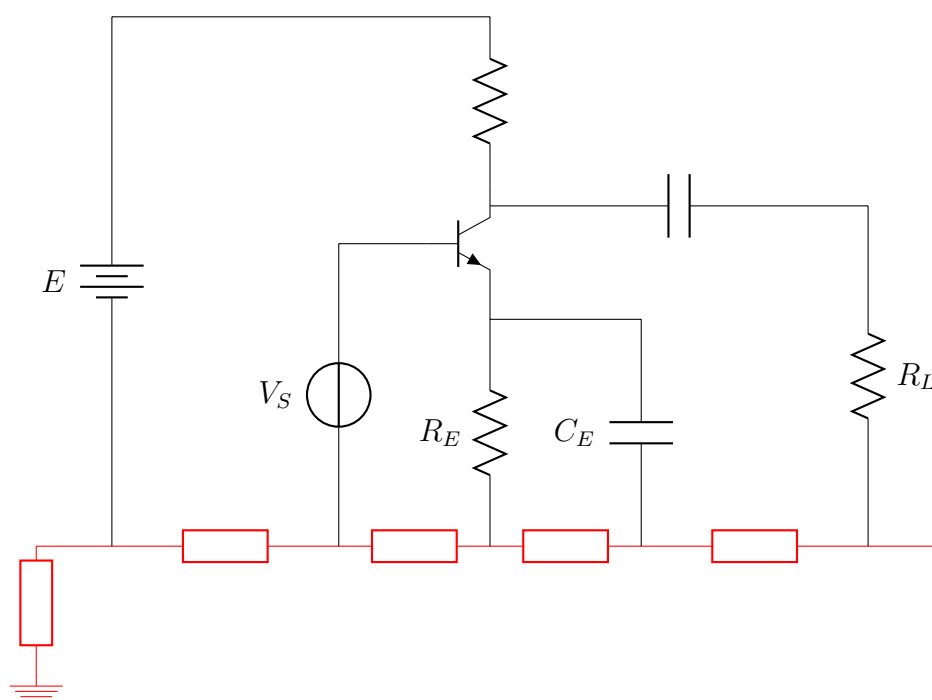


Figura 5.3: Impedenza del circuito di massa multipoint

Partizionamento di un sistema

Un utile accorgimento per i circuiti è quello di predisporre un circuito di massa separato per ognuno dei blocchi critici del circuito; questa tecnica, di largo impiego, viene denominata *partizionamento* del sistema; per attuarla è necessario individuare delle porzioni circuitali che possiedano caratteristiche sufficientemente comparabili sia per quanto riguarda l'ampiezza delle grandezze, sia per il loro contenuto spettrale: ad esempio, in un'apparecchiatura, lo stadio di alimentazione è sicuramente interessato da correnti maggiori rispetto alle singole parti circuitali; inoltre, gli stadi iniziali hanno esigenze diverse rispetto agli stadi finali.

5.1.3 Il piano di massa

La soluzione indubbiamente più sfruttata nella realizzazione di circuiti elettronici è costituita dal *piano di massa*, che è una struttura conduttrice che presenta la più bassa impedenza possibile, sulla quale il percorso di ritorno delle correnti è scelto dalle singole correnti: tale cammino è detto *percorso a minima impedenza*. L'impedenza di un piano conduttore è usualmente espressa nella forma: $Z = R + jX$, in $\frac{\Omega}{\square}$, se la distanza d fra due punti del piano metallico rispetta la relazione: $d < \frac{\lambda}{20}^3$; se il materiale è il rame, esso ha resistività $\rho = 17.2 \mu\Omega \cdot cm$, la resistenza del cubo di 1 cm di lato è pari a $17.2 \mu\Omega$; si consideri un parallelepipedo di rame con lato l pari a 1 cm e spessore t pari a 1 mm. La sua resistenza diventa:

$$R_{parallelepipedo} = \rho \frac{1 \text{ cm}}{1 \text{ cm} \cdot t} = \frac{\rho}{t} = \frac{17.2}{0.1} = 172 \mu\Omega \quad (5.1)$$

cioè 10 volte quella del cubo; notiamo che, se modifichiamo la dimensione del lato con un valore qualsiasi l , il valore di resistenza non cambia, poiché, dalla seconda legge di Ohm, si ha che:

$$R = \rho \frac{l}{l \cdot t} = \frac{\rho}{t} \quad (5.2)$$

L'indicazione dell'impedenza di un piano conduttore ha attinenza soprattutto all'aumentare della frequenza, poiché, se lo *spessore di penetrazione* è inferiore allo spessore del piano, la resistenza non è più in funzione dello spessore, bensì della *frequenza*; una relazione che permette di valutare questa grandezza è la seguente:

$$X = \frac{369 \sqrt{\mu_r \rho_r f}}{1 - e^{-\frac{t}{\delta}}} \frac{\mu\Omega}{\square} \quad (5.3)$$

dove permeabilità e resistività sono quelle specifiche del rame e δ rappresenta lo spessore di penetrazione; nel caso in cui lo spessore di penetrazione sia inferiore allo spessore t del piano di rame, la relazione (5.3) diventa:

$$X = 369 \sqrt{f_{MHz}} \frac{\mu\Omega}{\square} \quad (5.4)$$

Tale equazione è applicabile in un range di frequenza che vai 10 KHz (per spessori di 1 mm) a 1 MHz (per spessori dell'ordine degli 0.1 mm).

³cioè la distanza fra i due punti deve essere piccola rispetto alla *lunghezza d'onda* di interesse.

Presenza di fessure sul piano di massa

A livello ideale, un piano di massa dovrebbe essere una superficie continua di materiale conduttore; nella pratica però risulta essere spezzato in vari punti, al fine di consentire le interconnessioni fra i vari componenti, gli alloggiamenti per i connettori ed i componenti stessi, e così via; quando si posizionano i componenti risulta necessario evitare di creare delle interruzioni notevoli, dette *porzioni* di piano, creando delle connessioni che lo interrompano *in un numero limitato di punti*.

Dato il numero solitamente elevato di interconnessioni necessarie a realizzare un circuito, può venire al progettista la tentazione di sfruttare il piano di massa anche per le connessioni; per realizzare ciò, la quantità di rame asportata è solitamente risulta essere poca, però se la *geometria* dell'interconnessione è errata, si incorre nel rischio di creare due porzioni di piano, come appena segnalato. E' comprensibile, quindi, che la creazione di un circuito digitale non possa avvenire su due sole facce, a meno che questo non sia *relativamente semplice* e funzionante a frequenze *non elevate*; in tutti gli altri casi, è meglio procedere con una configurazione *multistrato*, ponendo le connessioni dei segnali che presentano un contenuto spettrale non troppo elevato sulle facce esterne, mentre nei piani interni predisporre due piani di massa (o almeno un piano di massa e uno per l'alimentazione) e, fra questi, un piano per le piste interessate da segnali a frequenza elevata.

Le soluzioni multistrato aumentano il costo di produzione del circuito, però vanno prese in seria considerazione per tutti quei sistemi digitali che operano ad alta frequenza poiché aiutano molto a risolvere i problemi di compatibilità.

Artifici per aumentare il piano di massa

Abbiamo detto che per diminuire l'impedenza di un piano di massa è necessario realizzare delle superfici quanto più estese possibile; è possibile sfruttare alcuni accorgimenti, a parte il costoso utilizzo di piani disposti in parallelo: è possibile che vi siano superfici estese fra le piste per il trasporto dei segnali le quali non hanno connessioni al loro interno: in tal caso è opportuno riempire queste aree con del rame, invece di lasciare lo spazio libero, ricordando che queste zone **devono** essere collegate al piano di massa con un numero di collegamenti elevato; va inoltre ricordato che, se ci si dimenticasse di connetterle al piano di riferimento,

si creerebbero delle *isole* a potenziale fluttuante, le quali possono peggiorare la situazione invece che migliorarla⁴.

5.1.4 Impiego di 4 o 6 strati nel PCB

Tenendo presente le normative sulla compatibilità elettromagnetica e le caratteristiche attuali dei circuiti integrati, è possibile dare una grossolana classificazione delle schede digitali⁵ in alta o bassa velocità.

Torna utile definire anche se un segnale è periodico o meno: un segnale si dice *periodico veloce* se la sua frequenza fondamentale è di qualche MHz; un esempio tipico è un segnale di clock; nei sistemi digitali sono anche presenti segnali che non sono periodici: vanno considerati ad alta frequenza se presentano intervalli di durata di almeno $10\ \mu\text{s}$ durante i quali sono sufficientemente ripetitivi, con una durata del singolo impulso dell'ordine dei 500 ns: un esempio di questa tipologia sono i segnali RAS⁶ e CAS⁷ impiegati per il controllo delle DRAM⁸.

Per evitare che le tracce che trasportano tali segnali possano essere fonte di disturbi è necessario posizionare tali tracce fra due piani con un potenziale costante a radiofrequenza (RF), con lo scopo di schermarle; ciò può avvenire predisponendo un piano di massa ed uno di alimentazione⁹; per le modalità di funzionamento dei sistemi digitali, tali piani sono collegati da vari condensatori di disaccoppiamento: si può dire che il potenziale di tali piani sia costante, quindi nullo ad alta frequenza.

Supponendo di avere 4 strati, le possibilità di utilizzo dei piani sono piuttosto ridotte (e dipendono molto dalla tipologia di segnali presenti nel circuito):

1. Disporre esternamente i piani e internamente i piani delle piste: in questo caso, anche se è rispettata la regola che i segnali devono essere infrapposti fra piani a potenziale RF fisso, possono sorgere inconvenienti dovuti a fenomeni di crosstalk fra i due piani interni.

⁴Le *islands* possono avere effetti molto negativi: molti programmi CAD per progettare schede elettroniche possiedono tools in grado di riconoscere ed eliminare queste aree flottanti di rame.

⁵PCB sta per "Printed Circuit Board", il circuito stampato.

⁶Row Address Strobe: è un segnale di comando per memorie dinamiche.

⁷Column Address Strobe: è un segnale di comando per memorie dinamiche.

⁸Dynamic Random Access Memories.

⁹Oppure, se il budget lo permette, di un doppio piano di massa.

2. Infrapporre i piani agli strati delle piste: sullo strato esterno vanno collocati tutte le piste dei segnali lenti, a differenza di quello interno, nel quale sono presenti tutte le piste dei segnali HF.

Notiamo che le soluzioni proposte non abbandonano *mai* l'infrapposizione dei segnali HF fra due piani a potenziale RF costante; adottando i criteri appena visti è possibile aumentare il numero degli strati a 8, 10 o anche a 12, collegando più piani di massa in parallelo e infrapponendoci i piani contenenti le piste di segnali *che presentano le stesse caratteristiche*.

Ulteriori accorgimenti

La tecnica di “chiudere” le piste che veicolano segnali ad alta frequenza è utile, ma non va dimenticato che i componenti possono essere collocati solamente sui piani esterni¹⁰ (“Top” e “Bottom” del PCB), che dovrà essere più breve possibile, poiché il tratto esterno non si avvale della stessa protezione di quello interno; è possibile calcolare la massima lunghezza di questo tratto di pista “esposto” nel seguente modo:¹¹ sapendo che nel vuoto la lunghezza d’onda risulta:

$$\lambda = \frac{v}{f} = \frac{3 \cdot 10^8}{10^9} = 30 \text{ cm} \quad (5.5)$$

Supponendo che la costante dielettrica relativa del materiale che compone la scheda sia 4, si ricava che λ a 1 GHz è pari a circa 15 cm; un’antenna ha la massima “efficienza” a circa $\frac{\lambda}{4}$, cioè circa a 4 cm: il tratto di pista “esposto” dovrà quindi avere una lunghezza inferiore ai 4 cm, al fine di attenuare il più possibile gli effetti di una *antenna non intenzionale*.

Le piste di guardia

Supponiamo che non sia possibile optare per una soluzione multistrato, oppure che in nello stesso piano di una scheda a due strati coesistano piste per segnali a bassa ed alta frequenza. Per attenuare i disturbi presenti in questi circuiti e limitare le conseguenze di una loro possibile irradiazione o ricezione, è possibile ricorrere alle *piste di guardia*: esse sono piste collegate a massa, costruite ai lati

¹⁰La connessione avviene attraverso in foro passante, chiamato in Inglese “hole”.

¹¹Il quale tiene conto che le norme impongono di verificare i disturbi irradiati fino alla frequenza di 1 GHz.

delle linee di segnale che vanno protette; l'effetto schermante di tali tracciati consente di ottenere degli ottimi miglioramenti.

Le alimentazioni

La progettazione dei piani per le alimentazioni richiede una cura analoga a quella utilizzata per i piani di massa, poiché i piani di alimentazione forniscono potenza a tutti i componenti presenti nel circuito.

Solitamente l'alimentazione alla scheda arriva tramite il connettore principale o su di un connettore separato e normalmente il circuito necessita di più alimentazioni distinte (nel caso del progetto qui esposto sono ben 8: 12V, 5V, 3.3V, 1.8V, 1.5V, 1.0V costante, 1.0V variabile, 0.9V); i sistemi digitali generano parecchi disturbi, che vanno confinati per evitare che si propaghino senza controllo: è quindi opportuno inserire all'ingresso dell'alimentazione un *filtro* che possiede il duplice scopo di bloccare i disturbi esterni ed impedire a quelli interni di propagarsi attraverso i conduttori di alimentazione. Solitamente questi filtri hanno una struttura a Π .

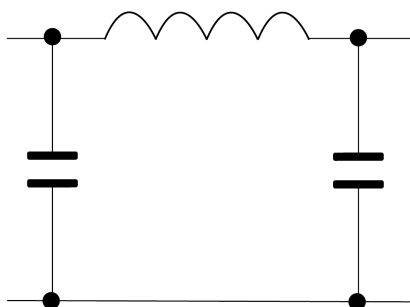


Figura 5.4: Schema circuitale di un filtro Π

Nel distribuire l'alimentazione così filtrata, si deve cercare di formare dei raggruppamenti quanto più uniformi è possibile: essi prendono il nome di *segmenti*, ed ogni filtro relativo viene detto *filtro di segmento*; ognuno di essi non deve contenere al suo interno un numero eccessivo di circuiti integrati: se ciò accade, probabilmente si renderà necessario attuare una separazione.

Fra gli accorgimenti di progettazione dei segmenti più importanti va tenuto presente che i segnali ad alta frequenza debbono compiere i tragitti più corti possibile; è quindi compito del progettista collocarli quanto più vicino possibile all'ingresso

dell'alimentazione; è inoltre importante la corrente dei componenti più sensibili ritorni all'ingresso attraversando porzioni di piano di massa interessate da correnti con intensità o $\frac{di}{dt}$ elevate.

Ultima considerazione, ma non meno importante, è che il filtro di segmento va collocato **nella posizione più vicina possibile all'ingresso dell'alimentazione** e non nella parte più lontana.

5.1.5 **Tecniche di protezione da sovratensioni e sovracorrenti**

Fenomeni quali sovratensioni o sovracorrenti fanno parte del normale funzionamento di un'apparecchiatura elettronica, però se non controllati possono comprometterne il funzionamento sia transitoriamente, sia permanentemente (provocando danneggiamenti); per attenuarne gli effetti è necessario predisporre degli opportuni accorgimenti di protezione.

Sorgenti di sovratensioni e sovracorrenti

Sovratensioni e sovracorrenti sono delle tipologie di disturbi di natura *impulsiva* che si sovrappongono alle grandezze nominali del circuito: sono caratterizzati da ampiezze notevoli, ma basse energie, a causa della loro durata limitata.

Le principali cause di sovratensioni e sovracorrenti sono:

- la chiusura ed apertura di interruttori: in un qualsiasi circuito in cui siano presenti elementi in grado di immagazzinare temporaneamente energia avvengono sovratensioni e sovracorrenti; è tipico riscontrare valori nell'ordine del kV per le tensioni e la fase più critica generalmente è rappresentata dall'apertura degli interruttori. Gli impulsi che avvengono in chiusura ed apertura vengono detti *burst* poiché si presentano a *pacchetti di impulsi*.
- le scariche atmosferiche: durante i temporali si manifestano scariche fra le nubi e terra con tensioni dell'ordine di milioni di Volt e correnti fino a 10^5 A; queste scariche provocano per induzione elettrostatica ed elettrodinamica disturbi sulle linee di distribuzione dell'energia elettrica, che si propagano su tutte le apparecchiature connesse: questi disturbi prendono il nome di *surge*, possiedono una energia associata superiore ai burst, presentano polarità sia positiva che negativa.
- le scariche elettrostatiche: l'importanza di questa classe di cause è aumentata sempre più con l'evoluzione delle tecnologie elettroniche poiché le apparecchiature sono sempre più vulnerabili alle scariche: è quindi motivato lo sforzo dei costruttori di proteggere i loro dispositivi contro le ESD.

Cenni di protezione nei circuiti integrati

Un accorgimento utilizzato a livello di circuiti integrati per evitare che i disturbi impulsivi di tensione provochino danni è sfruttare la *protezione mediante diodi collegati ai conduttori di alimentazione*. Nel collegamento che dall'esterno giunge all'ingresso del circuito integrato sono allocati due diodi di protezione connessi fra il conduttore di segnale V_{IN} , $+V_{CC}$ e $-V_{EE}$ ¹².

Durante il normale funzionamento del circuito è ragionevole supporre che il segnale di interesse V_{IN} rimanga sempre entro i limiti prefissati: in questa configurazione, entrambi i diodi sono *interdetti* e quindi la loro presenza è teoricamente ininfluenza nel funzionamento del circuito; invece, in presenza di un disturbo impulsivo nel circuito superiore in modulo al riferimento, uno dei due diodi entra in conduzione, limitando la tensione $-V_{IN}$ al massimo valore previsto in modulo (che corrisponde a V_{CC} oppure V_{EE}), addizionata della tensione di soglia del diodo conduttore.

La presenza dei diodi di protezione ha però delle conseguenze sul circuito integrato: un impulso di tensione, anche se viene efficacemente limitato entro V_{CC} e V_{EE} , può produrre false commutazioni di elementi di memoria interni, quali i flip flop; inoltre quando uno dei due diodi entra in conduzione, la corrente impulsiva di disturbo viene veicolata verso i circuiti di alimentazione o di massa, con conseguenze non sempre prevedibili: fortunatamente questo problema è risolvibile grazie all'*inserimento di un condensatore* fra $+V_{CC}$ e $-V_{EE}$, con lo scopo di stabilizzare il livello delle f.e.m.¹³ al verificarsi di disturbi impulsivi, immagazzinando o cedendo cariche elettriche in presenza del disturbo. Questi componenti di protezione solitamente hanno un'occupazione di area di silicio *confrontabile o superiore* a quelle aree occupate dai componenti funzionalmente utili: ciò è dovuto al fatto che devono essere in grado di sopportare tensioni e correnti impulsive piuttosto elevate.

¹²Nel caso di *alimentazione unipolare*, $-V_{EE}$ viene sostituita dal riferimento a potenziale nullo.

¹³Forze elettromotrici.

Conclusioni e sviluppi futuri

Nel corso dello svolgimento del progetto mi sono scontrato con molte problematiche, sia di elettronica, sia di informatica, sia di natura economica: la realizzazione di un sistema embedded completo richiede conoscenze ad ampio spettro, ho potuto comprendere quindi che è un lavoro che va svolto affidandosi alla conoscenza specifica di ognuno dei membri del gruppo di progetto; è stata un'esperienza di crescita, grazie alla quale ho aumentato il mio bagaglio culturale.

Il mio operato si è soffermato alla fase di *stesura degli schematici*: la naturale prosecuzione del lavoro porterà alla realizzazione del *layout* della board e alla produzione del *master*; dal master verrà costruito un *prototipo*, sul quale verrà effettuato un meticoloso *debug*, al fine di correggere gli errori commessi in fase di schematizzazione, di ottimizzare maggiormente il prodotto e di passare il *controllo EMI* (da svolgersi in camera anecoica¹⁴, presso un ente certificatore).

Un volta compiuto il debug e prodotto il software finale, il dispositivo sarà pronto per la produzione in serie.

¹⁴Le camere elettromagneticamente anecoiche vengono sfruttate per condurre i test di *compatibilità elettromagnetica*, previsti dalla normativa vigente a livello mondiale: qualsiasi dispositivo elettrico prima di essere immesso sul mercato deve, infatti, essere provato in modo da non provocare interferenze con altri apparecchi elettrici nelle vicinanze e al contempo, non subire i disturbi che queste stesse apparecchiature diffondono.

Ringraziamenti

Ringraziamenti

La stesura di un lavoro di tesi è un'opera che dà molta soddisfazione e contribuisce a migliorare e approfondire la propria conoscenza nell'ambito sul quale essa si basa; su questo lavoro compare il mio nome, tuttavia non sarebbe stato possibile redarlo senza il supporto del Professor Neviani, al quale rivolgo i miei ringraziamenti, sia per il sostegno che mi ha fornito sui due progetti di tesi della mia carriera universitaria, sia per i contenuti che è stato in grado di trasmettermi durante i corsi da Lui tenuti.

Un ringraziamento va anche all'Ingegnere Tellatin di Si14, per avermi affidato la realizzazione di questo lavoro e tutti gli strumenti aziendali per portarla a compimento; ringrazio anche gli ingegneri Guseo, Zuccollo, Bianconi, Baltieri e Burelli per le conoscenze che mi hanno pazientemente trasmesso e per il tempo dedicati. Desidero ringraziare anche Alberto B., Thomas, Fabio C. e Damiano per avermi sempre offerto una visione diversa dalla mia, dandomi modo di comprendere le problematiche da un'altro punto di vista.

Non posso che ringraziare i miei genitori, una presenza costante ed intensa nella mia vita, sempre pronti a soddisfare le mie esigenze al fine di raggiungere i risultati che mi prefiggo; un grosso debito di gratitudine ce l'ho con i miei nonni, che tanto mi vogliono bene e che, con pazienza, hanno atteso questo momento.

Un dolce ringraziamento va anche a Rossella.. i suoi occhi non riescono a celare la bontà del suo animo; la sua vicinanza ed il suo conforto morale per me sono stati determinanti.

Il mio pensiero va anche ad Alberto Michielan e Luca Parmesan, compagni di non so quante giornate fra il bus, le aule e gli scaffali della biblioteca: sono stati giorni di lavoro, ma anche il contesto in cui si è sviluppata una bella amicizia.

Concludo ringraziando Gabriele, l'Amico per la vita che molti vorrebbero, ma pochi possono vantarsi di avere: per me la sua costanza nel lavoro è sempre stata d'esempio, il dialogo che instauriamo è un continuo momento di crescita.

Elenco delle figure

1	Un esempio di HMI	1
2	Un esempio di HMI privo dello chassis, con Android OS	2
1.1	Esempi di sistemi embedded: sono tutti diversi fra loro	4
2.1	Schema a blocchi del sistema	17
2.2	Pandaboard Evaluation Board	19
2.3	Le memorie RAM di tipo Package-on-Package	20
2.4	i.MX536 Application Processor Evaluation Kit	22
2.5	Schema a blocchi funzionale del processore AM3892	25
2.6	AM3892 Application Processor Evaluation Module	29
2.7	Struttura di un dominio di alimentazione	32
2.8	Domini delle tensioni di alimentazione	35
2.9	Requisiti per l'avvio sequenziale delle tensioni di alimentazione	36
2.10	Il clock manager	38
2.11	Il clock a 27 MHz	39
2.12	Schema a blocchi del blocco dei Timer	40
2.13	Schema a blocchi del blocco dei Timer	41
2.14	Schema a blocchi del bus I ² C	42
2.15	Gli endpoint USB	43
2.16	Schema a blocchi del sottosistema USB dell'AM3892	45
2.17	Schema a blocchi del charge pump TPS2052B	48
2.18	Schema a blocchi del sottosistema audio dell'AM3892	49
2.19	Schema a blocchi del codec audio TLV320AIC3106	50
2.20	Configurazione tipica del codec audio TLV320AIC3106	52
2.21	Schema del controllore SecureDigital	53
2.22	RS-232: connessione punto-a-punto	55

2.23	Un esempio di trasmissione tramite RS-232	56
2.24	Confronto fra LVDS ed altri tipi di segnale noti	57
2.25	Rappresentazione tipica del segnale LVDS nel canale	57
2.26	Schematizzazione di un Driver ed un Receiver LVDS	58
2.27	Schema a blocchi del SERDES LVDS	60
2.28	Il ricevitore montato sul display touchscreen	61
3.1	Schema elettrico semplificato di un convertitore buck	65
3.2	Schema funzionale di un regolatore lineare	68
3.3	Regolatore Standard (NPN)	69
3.4	Regolatore LDO	70
3.5	Regolatore quasi-LDO	71
3.6	Circuito di limitazione a corrente costante	73
3.7	Circuito di limitazione a corrente costante	73
3.8	Influenza della ESR sulla funzione di trasferimento	74
3.9	Sorgenti di alimentazione e correnti erogabili	76
3.10	Consumi del sistema osservati tramite porta seriale	77
3.11	Sequenza di power-up del sistema del powermanager TPS659112	77
3.12	Schema semplificato e retta di carico del TPS54620	78
5.1	Schema di principio di un amplificatore	106
5.2	Schema di massa di tipo multipoint	107
5.3	Impedenza del circuito di massa multipoint	108
5.4	Schema circuitale di un filtro II	114

Bibliografia

- [1] John Catsoulis, *Designing Embedded Hardware*, O'Reilly Media, 2nd Edition, June 2009.
- [2] Jack Ganssle, *Embedded Systems - World Class Designs*, Elsevier, Newnes Press, 4th Edition, 2010.
- [3] *Embedded System Design Specifications*, Si14 Embedded Solutions, Padova, 2nd Revision, 2011.
- [4] *AM389x Sitara ARM Microprocessors*, Texas Instruments, 1st Edition, October 2010.
- [5] *AM389x Sitara ARM Microprocessors Technical Reference Manual*, Texas Instruments, 1st Edition, January 2011.
- [6] *TLV320AIC3106: Low-power Stereo Audio Codec for Portable Audio/Telephony*, Texas Instruments, December 2008.
- [7] *SN65LVDS93: LVDS SerDes Transmitter*, Texas Instruments, May 2009.
- [8] *HSD100IFW1-A00: 10.1" TFT liquid crystal display*, Texas Instruments, May 2009.
- [9] Ned Mohan, Tore M.Undeland, William P. Robbins, *Power Electronics*, Hoepli, Milano 3rd Edition, 2005.
- [10] Chester Simpson, *Linear and Switching Voltage Regulator Fundamentals*, National Semiconductors, 2007.
- [11] *TPS659112: Integrated Power Management Unit*, Texas Instruments, February 2011.

BIBLIOGRAFIA

- [12] *TPS54620: 4.5V to 17V Input, 6A, Synchronous Step Down Converter*, Texas Instruments, October 2010.
- [13] Matteo Bertocco, Alessandro Sona, *Manuale di compatibilità elettromagnetica*, 2^a Edizione, 2010.

