

UNIVERSITÀ DEGLI STUDI DI PADOVA

Facoltà di Ingegneria

Corso di Laurea in Ingegneria Elettronica



Relazione di stage

Progettazione di un driver per schermo LCD

Tutor universitario/Relatore:

Prof.

Daniele VOGRIG

Tutor aziendale

Ing.

Massimo BORELLA

Laureando

Nicola BOLZAN

Anno accademico 2011/2012

Sommario

Grazie agli sviluppi tecnologici a cui abbiamo assistito in questi anni, si sono resi disponibili schermi LCD una volta utilizzabili solo in applicazioni costose. Per questo motivo è oggi possibile utilizzare questo tipo di schermi in applicazioni industriali una volta precluse, per motivi economici, all'utilizzo di tali dispositivi.

Il problema che però ci si trova ad affrontare nell'utilizzo di questi componenti è la scarsa reperibilità di sistemi di controllo che permettono il pilotaggio di questi display da parte di un microcontrollore.

In questa tesi si affronterà quindi la progettazione di una scheda driver in grado di pilotare un display monografico con risoluzione 320 per 240 pixel. La scheda è stata progettata per essere pilotata da un microcontrollore il quale la gestisce come fosse una memoria grafica in cui va a salvare i dati dell'immagine da visualizzare, demandando alla scheda stessa la gestione dei segnali necessari per il corretto pilotaggio dello schermo LCD. Per fare ciò la scheda si basa sull'utilizzo di due CPLD per quanto riguarda la gestione dei vari componenti e la generazione dei segnali di controllo.

Inoltre, la scheda può essere inserita in un sistema in cui siano presenti più dispositivi di I/O in cui la gestione dei vari dispositivi avviene tramite la tecnica di *memory mapping*.

Indice

Capitolo 1: Introduzione	7
1.1 L'azienda	7
Capitolo 2: Panoramica del tirocinio	9
2.1 Il dispositivo di partenza	9
2.2 Obiettivi del progetto	9
2.4 Specifiche di progetto	10
Capitolo 3: Progettazione della scheda	11
3.1 Sviluppo delle specifiche	11
3.2 L'hardware utilizzato	11
3.2.1 Lo schermo LCD	11
3.2.1.1 Display serie 240320DP	13
-Caratteristiche fisiche	13
-Segnali di controllo	14
-Tensioni di funzionamento	15
3.2.2 Alimentazione	15
3.2.2.1 MCP1825	15
3.2.2.2 MC34063	16
3.2.3 Il modulo RAM	17
3.2.4 Latch	18
3.2.5 CPLD	18
3.2.5.1 Tecnologie di realizzazione	19
3.2.5.2 Implementazione di un circuito	22
-XILINX ISE WebPack	23
3.3 Principio di funzionamento della scheda	23
3.3.1 Interfaccia verso il microcontrollore	25
3.3.1.1 Scrittura dell'immagine nella memoria	25
3.3.1.2 Lettura dell'immagine in memoria	27
3.3.2 Interfaccia verso il display	28
3.3.2.1 Visualizzazione sullo schermo LCD dell'immagine	29

3.4 Programmazione della CPLD	30
3.4.1 CPLD1	30
3.4.2 CPLD2	35
3.5 Lo schematico	43
3.5.1 Individuazione blocchi funzionali	43
3.5.2 Stesura dello schematico	45
3.5.3 Individuazione errori	51
Capitolo 4: Realizzazione e testing del prototipo	53
4.1 Realizzazione prototipo	53
4.1.1 Realizzazione prototipo alimentazioni	53
4.1.2 Il master	55
4.1.2.1 Estrazione della netlist e lista componenti	55
4.1.2.2 Individuazione footprint	55
4.1.2.3 Organizzazione dei componenti nella scheda	56
4.1.2.4 Stesura master	57
4.1.2.5 Stampa del prototipo	58
Bibliografia	59

Capitolo 1: Introduzione

In questa tesi si andrà ad illustrare il progetto di una scheda di controllo per un driver LCD monografico sviluppata durante un tirocinio in azienda. Il progetto si basa sull'utilizzo di CPLD per il controllo di una scheda driver che va a pilotare un display LCD con risoluzione 320 per 240 pixel.

Il tirocinio e l'azienda

Il tirocinio è stato svolto presso l'azienda E.SHARP [1], azienda che si occupa della progettazione e realizzazione di sistemi elettronici di controllo per macchine industriali basati su microcontrollore.

Per il controllo delle apparecchiature realizzate vengono utilizzati vari tipi di display LCD, a seconda dell'uso per il quale è progettata la scheda a microcontrollore.

Volendo realizzare un sistema di gestione per un display LCD con risoluzione maggiore rispetto a quella dei display abitualmente utilizzati in azienda, si è deciso di demandare la progettazione ad uno studente universitario che lo avrebbe sviluppato nell'ambito di un tirocinio in azienda. Si è deciso di ricorrere ad uno studente per il fatto che il progetto non necessitava di essere completato in un tempo breve ma voleva essere più uno studio nell'ottica di un futuro utilizzo in azienda di display con caratteristiche più performanti rispetto ai dispositivi già utilizzati.

Capitolo 2: Panoramica del tirocinio

2.1 Il dispositivo di partenza

Nell'azienda in cui si è svolto il tirocinio vengono utilizzati vari tipi di display. In particolare viene utilizzato un display LCD con risoluzione 128 per 240 pixel prodotto dalla Emerging Display [15]. Questo display è di tipo LCD grafico, e viene fornito con a bordo il controller T6963C prodotto dalla Toshiba. Questo controller permette una gestione del display sia come display grafico che come display a caratteri, rendendo disponibile un set di caratteri memorizzati nella ROM del controller.

Se da una parte il controller è facilmente utilizzabile, esso presenta però anche delle limitazioni: infatti, a causa della tecnica di pilotaggio, permette un refresh completo dello schermo con una frequenza massima pari a 12 frame al secondo, frequenza troppo bassa per permettere di realizzare semplici animazioni.

Per ovviare a questo problema e produrre un sistema di controllo che permettesse di controllare un display più performante si è deciso di realizzare un nuovo sistema di controllo per un display LCD grafico la cui descrizione sarà illustrata di seguito.

2.2 Obiettivi del progetto

Nella realizzazione del progetto si è voluta mantenere una certa similarità con i sistemi di controllo dei display già utilizzati in azienda, pur dovendo apportare sostanziali modifiche quali il tipo di connessione o la politica di controllo. Inoltre, per rendere possibile l'utilizzo del display in sistemi simili a quelli in cui è già presente il display meno risoluto, si sono volute mantenere dimensioni esterne del sistema display-controller compatibili con quelle del display già utilizzato.

Per questo motivo la scelta del componente è ricaduta su un display con risoluzione 320x per 240 pixel monocromatico prodotto dalla *Displaytech* [2].

Differentemente dal precedente, però, questo display non dispone di un driver interno che ne curi il refresh, si è reso quindi necessario lo sviluppo di una scheda che realizzasse l'aggiornamento dello schermo, anche a causa del costo dei dispositivi di pilotaggio presenti sul mercato e della loro scarsa reperibilità.

Per fare ciò, tramite la consultazione del datasheet del display stesso e di progetti simili reperiti in internet, in azienda si è scelto di sviluppare una scheda in cui la generazione dei

segnali di pilotaggio dello schermo fosse realizzata da una CPLD. Ora, essendo il primo progetto dell'azienda basato su una CPLD e dato che non era necessario un completamento a breve termine del progetto, si è deciso di ricorrere ad uno studente universitario offrendo la possibilità di un tirocino in azienda durante il quale sviluppare il progetto.

2.3 Specifiche di progetto

Il dispositivo che si vuole andare a realizzare deve soddisfare alcune specifiche imposte dall'azienda.

La prima è la capacità della scheda di compiere un refresh completo dello schermo con una frequenza di 50 Hz, in maniera che non siano distinguibili sfarfallii o altri problemi di visione. Inoltre la scheda deve poter interfacciarsi con microcontrollori che possono funzionare a 3.3 V come a microcontrollori alimentati a 5 V. La compatibilità con vari tipi di microcontrollori deve essere garantita anche per quanto riguarda l'architettura degli stessi, che può essere a 8, 16 o 32 bit.

Altra specifica riguarda le dimensioni, sia dello schermo che della scheda driver, che devono essere simili con quelle dello schermo con risoluzione massima già utilizzato in azienda.

La scheda driver deve inoltre poter essere inserita in un sistema di controllo in cui sono presenti anche altri dispositivi di input/output, e quindi deve soddisfare i criteri per essere pilotata in *memory mapped*.

Infine, essendo l'obiettivo del progetto la realizzazione di un prototipo, è necessaria la scelta dei componenti in base alla disponibilità in azienda e optare per tecnologie di montaggio realizzabili a mano, al fine di poter realizzare il prototipo senza l'ausilio ditte esterne, a meno della stampa del PCB.

Capitolo 3: Progettazione della scheda

Verranno ora analizzate le varie fasi che hanno portato alla progettazione della scheda driver.

3.1 Sviluppo delle specifiche

L'idea fondamentale del progetto è il realizzare una scheda che possa essere vista dal microcontrollore come una semplice memoria grafica su cui andare a scrivere l'immagine che si vuole visualizzare, demandando la generazione dei segnali di controllo e la gestione dei componenti alla CPLD interna alla scheda. In questo modo si sono individuati i componenti principali da utilizzare, e cioè:

- La CPLD, necessaria per la generazione dei segnali di controllo;
- Una memoria RAM statica, su cui il microcontrollore possa scrivere l'immagine. Si è scelto l'uso di una memoria statica per non dover realizzare anche il refresh della memoria stessa, dato che i tempi di accesso di una memoria statica sono comunque compatibili con le tempistiche richieste dal progetto;
- Una serie di buffer/latch, che permettano di separare i bus interni alla scheda con quello in comune al microcontrollore.

Viene di seguito fornito una rapida descrizione dell'hardware utilizzato, soffermandosi sul componente principale della scheda, la CPLD.

3.2 L'hardware utilizzato

3.2.1 Lo schermo LCD

I primi display a cristalli liquidi vennero alla luce nel 1968 presso il Centro Ricerche “David Sarnoff” negli Stati Uniti.

Nel corso degli anni, la tecnologia si è evoluta notevolmente e gli LCD (Liquid Crystal Display) sono principalmente applicati nei seguenti settori:

- monitoria;
- telefonia cellulare;
- strumentazione elettronica;
- orologeria.

Il funzionamento di ogni LCD è basato sulle proprietà di particolari sostanze denominate cristalli liquidi.

Tale liquido è intrappolato fra due superfici vetrose provviste di numerosi contatti elettrici con i quali poter applicare un campo elettrico al liquido contenuto. Ogni contatto elettrico comanda una piccola porzione del pannello identificabile come un pixel.

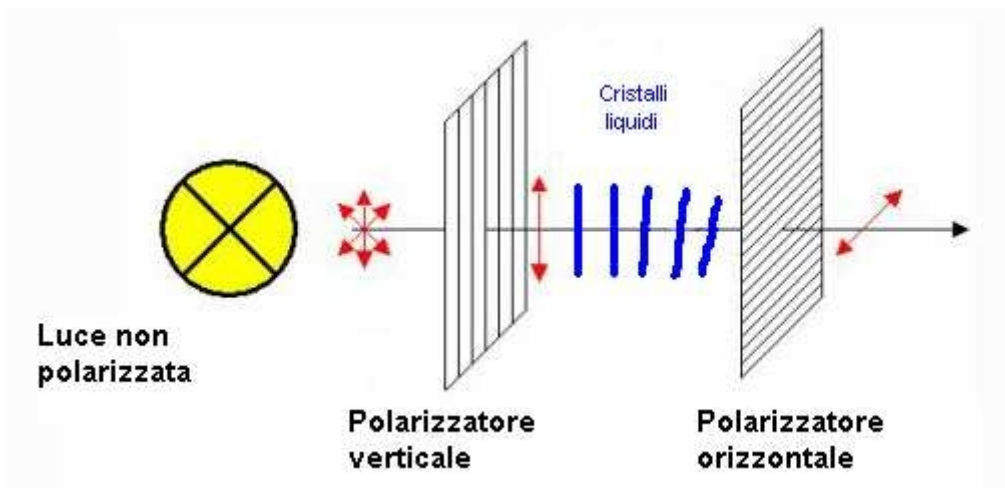
Sulle facce esterne dei pannelli vetrosi sono collocati due filtri polarizzatori disposti su assi perpendicolari tra loro. La particolarità naturale dei cristalli liquidi è torcere di 90° la polarizzazione della luce che arriva da uno dei due filtri, permettendole quindi di attraversare il secondo polarizzatore.

In assenza di campo elettrico, la luce può passare attraverso l'intera struttura e, trascurando la porzione di luce assorbita dai polarizzatori, l'apparecchio risulta trasparente. In presenza di un campo elettrico ortogonale ai piani delle due lastre di vetro, invece, le molecole del liquido si allineano parallelamente al campo, limitando la rotazione della luce entrante. Se i cristalli sono completamente allineati con il campo elettrico, la luce che vi passa attraverso è polarizzata perpendicolarmente al secondo polarizzatore e viene quindi bloccata del tutto facendo apparire il pixel non illuminato. Controllando la torsione dei cristalli liquidi in ogni pixel, tramite un terzo filtro, si è in grado di regolare quanta luce far passare. Si noti che in questo modo un pixel guasto apparirà sempre illuminato. Nella realtà alcune tipologie di pannelli funzionano all'opposto, ossia sono trasparenti se accesi ed opachi se spenti. In tal caso un pixel danneggiato risulta sempre opaco.

Un LCD non emette luce: ecco il motivo per cui questi schermi necessitano della retroilluminazione (back light). La luce emessa da questo sistema attraversa il cristallo liquido e vi esce colorata dal sistema di filtraggio.

L'attivazione di ogni singolo punto del display viene comandata da un meccanismo di indicizzazione "riga per colonna", per cui ogni pixel si attiva quando passa corrente in entrambi gli elettrodi (anteriore e posteriore) che lo riguardano.

Nel display TFT i singoli pixel sono attivati da un apposito transistor. Quindi non è più necessario porre davanti al video una serie di elettrodi: è sufficiente la presenza di un'unica lastra trasparente da impiegare come massa.



Principio di funzionamento di un cristallo liquido

Lo schermo LCD usato in questo progetto è un display monografico retroilluminato a LED con risoluzione 320 per 240 pixel della serie 240320DP prodotto dalla *Displaytech*.

3.2.1.1 Display serie 240320DP

Questo display è stato scelto per la sua facile reperibilità e per il suo costo contenuto, oltre che per le sue dimensioni, compatibili con le specifiche di progetto.

Caratteristiche fisiche

Questo schermo LCD è un display LCD STN/FSTN grafico monocromatico con risoluzione 320 per 240 pixel [3]. Il modulo LCD ha dimensioni 70x89x8 mm con un'area di visualizzazione di 61x80 mm, dimensioni simili a quelle del display già utilizzato in azienda. La retroilluminazione è a led di colore bianco.

Questo display è caratterizzato da una cornice in acciaio su cui sono praticati i 4 fori di fissaggio. Per la connessione con il circuito di pilotaggio, il display è fornito di un cavo flat piatto a 24 vie. L'alimentazione è compresa nei 24 connettori del cavo flat.

Per quanto riguarda le tensioni di alimentazione, devono essere forniti al display 3 livelli di tensione:

- V_{ee}: tensione che determina il contrasto, tensione pari a 26 V.
- V_{dd}: tensione di alimentazione della logica di controllo dell'LCD, cioè della logica che comanda il pilotaggio dei singoli pixel. Questa tensione è pari a 5 V.

- A: tensione di alimentazione dei LED di retro-illuminazione, pari a 3.3 V.

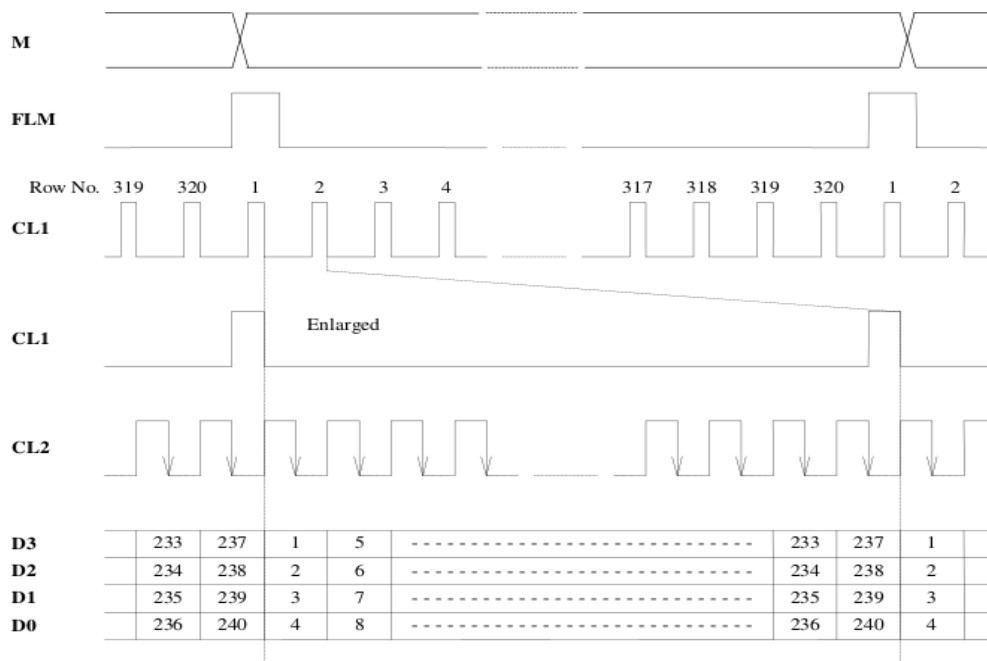
Il display presenta al suo interno un pilotaggio righe/colonne con cui è possibile accedere sequenzialmente ad ogni pixel dello schermo. La selezione dei pixel segue un metodo di pilotaggio che verrà descritto nella prossima sezione.

Segnali di controllo

Per un corretto pilotaggio del dispositivo è necessaria la generazione di diversi segnali, correttamente sincronizzati. I principali sono:

- M: segnale ad onda quadra che determina la scrittura di ogni frame. Ad ogni semiperiodo corrisponde un refresh completo dello schermo. Volendo un refresh ogni 20 ms (50 Hz) questo segnale deve avere una frequenza di 25 Hz.
- FLM : (frame signal) segnale che va a livello logico alto ogniqualvolta si inizia un nuovo refresh. Il segnale deve essere portato a livello logico alto dopo la scrittura dell'ultimo pixel e riportato a livello logico basso prima della scrittura del primo pixel.
- CL1: segnale che indica la riga su cui si va a scrivere. Ad ogni impulso, il contatore interno della riga viene incrementato di uno.
- CL2: segnale clock di scrittura. Dato che il display è monografico e che il bus dati del display è a 4 bit, ad ogni impulso di questo segnale vengono scritti sul display 4 pixel, perciò ad ogni impulso il contatore di colonna viene incrementato di 4 unità.

Viene di seguito riportato il grafico temporale dei segnali di controllo:



Come si nota, oltre ai segnali di controllo frame (FLM ed M), è necessario una temporizzazione precisa per quanto riguarda i segnali di clock riga e clock colonna. In particolare, i dati che vengono letti in ingresso dall'LCD devono essere disponibili prima che si verifichi il fronte di discesa del clock colonna (CL2) e possono essere aggiornati tra un fronte e l'altro. Si noti che, essendo monografico il display, è possibile dare in ingresso il valore di 4 pixel contemporaneamente.

Tensioni di funzionamento

Per il corretto funzionamento dei dispositivi presenti nella scheda sono necessarie 3 tensioni: una tensione a 3.3 V per l'alimentazione degli integrati (CPLD e RAM) e per l'alimentazione della retroilluminazione del display, una tensione a 5V per l'alimentazione dei latch di uscita e per la logica di controllo interna all'LCD e una tensione a 26 V necessaria come tensione di riferimento per il contrasto dell'LCD.

Ora, dato che nella scheda driver viene data in ingresso una tensione di alimentazione a 5 V, si è resa necessaria la realizzazione nella scheda delle tensioni 3.3 V e 26 V, la prima utilizzando l'integrato MCP1825 e la seconda utilizzando l'integrato MC34063.

3.2.2 Alimentazione

Essendo necessarie, per il corretto funzionamento della scheda, varie alimentazioni, esse sono state ricavate dall'alimentazione della scheda (5 V) e con l'utilizzo di componenti dedicati (+3.3 V e +26 V). Per ricavare queste alimentazioni sono stati utilizzati gli integrati MCP1825 e MC34063, rispettivamente per i 3.3 V e per i 26 V.

3.2.2.1 MCP1825

Questo integrato è un regolatore lineare di tensione a basso dropout [4], cioè riesce a dare una tensione in uscita minore della tensione che riceve in ingresso senza che le due tensioni abbiano una differenza significativa. Questo integrato può alimentare carichi fino a 500 mA. Essendo fornito con package TO-220, è possibile anche l'applicazione di dissipatori di calore qualora ci fossero problemi di potenza dissipata.

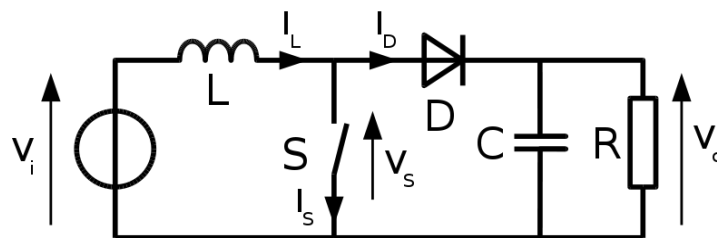
Dato che la retroilluminazione dell'LCD può assorbire un massimo di 360 mA, si è deciso di utilizzare 2 regolatori di tensione MCP1825, uno per l'alimentazione dei componenti della

scheda ed uno per i led di retroilluminazione. L'integrato non necessita di nessun componente esterno, ad esclusione di due condensatori tra alimentazione e massa necessari per stabilizzare la tensione in ingresso e quella in uscita.

3.2.2.2 MC34063

Questo integrato, prodotto dalla ST Microelectronics, è uno *step-up converter* [5] che realizza una tensione variabile in uscita partendo da una data tensione in ingresso. Per fare ciò, questo integrato realizza uno step-up sfruttando le proprietà di un induttore.

Uno step-up converter è un circuito che permette di innalzare una tensione ad un valore predefinito sfruttando le caratteristiche di un induttore.



Step-up converter: principio di funzionamento.

I componenti principali di un circuito step-up sono un induttore, un diodo, un condensatore ed uno switch. Il principio di funzionamento può essere diviso in due parti: la prima quando lo switch è chiuso e la seconda quando lo switch è aperto. Quando lo switch è chiuso, nell'induttore comincia a fluire della corrente. Una volta aperto, l'energia immagazzinata nella prima fase dall'induttore viene a poter essere scaricata solo attraverso il diodo, il quale carica il condensatore ai cui capi si viene a realizzare la tensione desiderata.

Su questo principio di funzionamento si basa l'integrato MC34063. Questo integrato presenta al suo interno gli elementi base necessari per la realizzazione di un convertitore step-up. Sono infatti presenti un oscillatore, uno switch di potenza e un comparatore. L'induttore e il condensatore necessario alla realizzazione dello step-up devono essere inseriti esternamente, con valori determinati dalle prestazioni, in termine di corrente di uscita, che si vogliono ottenere. L'integrato andrà a controllare l'apertura e la chiusura dello switch interno in maniera tale da mantenere la propria uscita al valore di tensione determinato dal partitore resistivo il cui valore viene riportato all'apposito ingresso di riferimento. Per dimensionare i componenti, ci

si è basati sulle indicazioni presenti nel datasheet del componente. Il dimensionamento dei componenti è riportato nella sezione dello svolgimento del progetto.

3.2.3 Il modulo RAM

In ambito elettronico la memoria è un dispositivo destinato a conservare informazioni per un certo periodo di tempo.

Esistono vari tipi di memoria, ma i principali sono:

- ROM: Read Only Memory: è una memoria permanente, cioè ha un contenuto fisso che non può essere cancellato ed inoltre non è volatile;
- EPROM :Erasable Programmable Read Only Memory: è una memoria in sola lettura, ma ha la particolarità di poter essere cancellata in particolari condizioni.
- RAM: Random Access Memory: è una memoria caratterizzata dal permettere l'accesso diretto a qualunque indirizzo di memoria con lo stesso tempo di accesso.

La memoria utilizzata in questo dispositivo è di tipo Static RAM, cioè una memoria RAM che non necessita di refresh essendo composta da dei latch, i quali una volta scritti non hanno bisogno di essere rinfrescati ciclicamente, cosa che non è verificata nelle Dinamic RAM, dove la memoria è realizzata tramite dei condensatori che però presentano delle perdite, per cui il valore in essi memorizzato deve essere ciclicamente ripristinato da un apposito circuito che si occupa del refresh.

Si è ricorsi all'utilizzo di una memoria statica perché, non avendo necessità di velocità molto elevate proprie delle RAM dinamiche, si è preferito ricorrere ad un tipo di memoria più facilmente gestibile, e che non necessitasse di un refresh continuo.

Volendo pilotare un LCD con risoluzione 320 x 240 pixel, sapendo che ogni pixel necessita di un bit, avremo bisogno di 9600 byte di memoria RAM per ogni immagine. Dato che si vuole mantenere una velocità di refresh di 50 Khz, è necessario scegliere un tipo di RAM con tempi di accesso adeguati. In particolare la memoria deve essere in grado di rendere possibile la lettura in un tempo massimo pari a 20 μ s (vedi sezione dedicata alla CPLD2). Inoltre, volendo avere a disposizione la memoria necessaria a salvare più immagini, e considerati gli integrati presenti in azienda, si è scelta una RAM statica da 128 KB prodotta dalla Cypress Semiconductor con package SOJ [6].

Questo tipo di memoria è una memoria RAM statica a tecnologia CMOS con tempo di accesso di 10 ns. La gestione delle operazioni di lettura e scrittura sono molto semplici, dato che il componente dispone dei 2 pin di READ e WRITE con i quali è possibile selezionare la modalità di funzionamento. Questo integrato è alimentato a 3.3 V.

3.2.4 Latch

Il latch è un circuito elettronico bistabile capace di memorizzare un bit di informazione.

In questo progetto essendo i latch utilizzati come interfaccia verso la MCU, e dovendo rispettare la specifica che essa possa operare a varie tensioni, si è dovuto scegliere un integrato che, pur operando in uscita a 3.3 V, fosse 5 V tolerant, cioè accettasse in ingresso anche segnali con soglie TTL. Inoltre, essendo le uscite dei latch di ingresso collegate al buffer dati e al buffer indirizzi della scheda, buffer in comune ad altri dispositivi, per evitare fenomeni di bus contention è necessario che le uscite dei buffer possano essere di tipo tri-state.

In base a queste specifiche la scelta è ricaduta sull'integrato 74LVC374, che presenta al suo interno 8 latch D con tensione tollerata in ingresso pari a 5V. In questo integrato è presente un pin di output enable (OE) che abilita le uscite dei latch quando è attivo, lasciando l'uscita in alta impedenza quando disabilitato. Per la memorizzazione dei valori posti in ingresso si utilizza invece il pin di Latch Enable (LE). Per rendere compatibile le tensioni di uscita con gli altri componenti della scheda, l'integrato deve essere alimentato a 3.3 V.

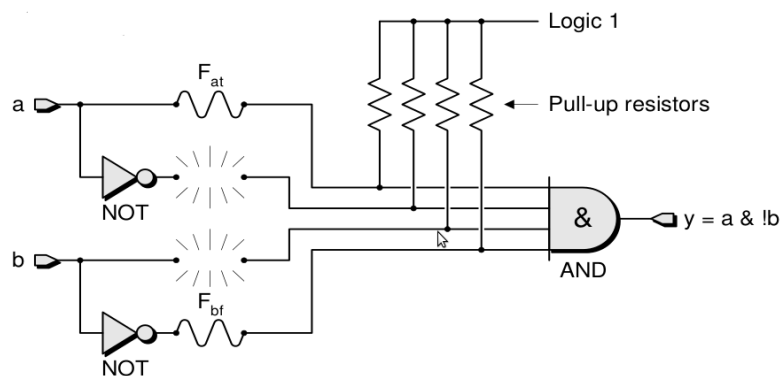
3.2.5 CPLD

Una CPLD (acronimo di Complex Programmable Logic Device), è un dispositivo programmabile con cui è possibile simulare un circuito logico [8]. Questo dispositivo è l'evoluzione dei PLD, un dispositivo la cui effettiva funzione logica viene stabilita attraverso una procedura di programmazione. I primi PLD (detti anche Simple PLD, SPLD) sono apparsi attorno al 1970 e consentivano di realizzare un insieme di funzioni combinatorie relativamente semplici. Con lo sviluppo della tecnologia CMOS, che ha reso possibile raggiungere la densità di integrazione e il basso consumo richiesto in un dispositivo simile, nel 1984 sono apparsi i primi CPLD, che in linea di principio consistevano in una serie di SPLD collegate tra di loro attraverso una matrice di connessioni programmabili.

3.2.5.1 Tecnologie di realizzazione

Per programmare queste connessioni si sono utilizzati negli anni vari metodi, di cui vengono elencati di seguito i principali:

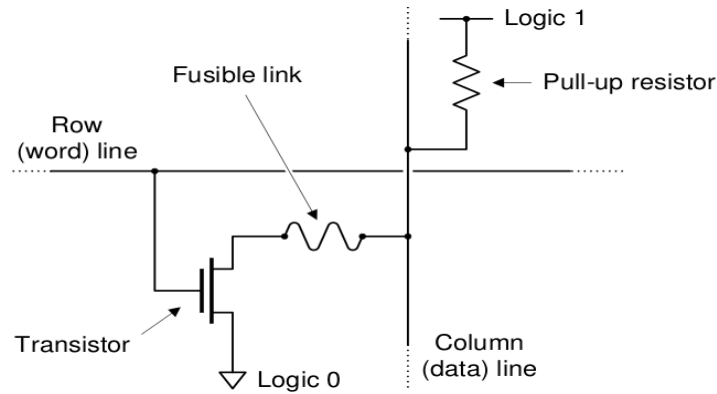
- Meccanismo di programmazione a fusibili, in cui il contatto o meno viene determinato dall'integrità del fusibile che realizza il contatto. Questo è stato il primo modo di realizzazione di matrici programmabili. Inizialmente tutte le connessioni sono presenti. In un secondo tempo tramite l'applicazione di una tensione elevata vengono “bruciati” i fusibili necessari alla realizzazione della funzione voluta.



Programmazione a fusibili

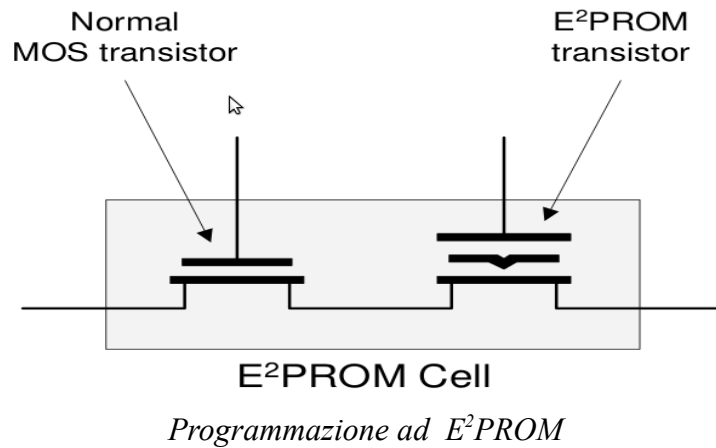
- Meccanismo di programmazione ad antifusibili. Questo metodo si basa sul principio complementare rispetto al precedente: inizialmente i contatti sono delle resistenze di valore elevato. Tramite l'applicazione di una tensione elevata, questi antifusibili vengono trasformati in connettori a bassa resistenza. Un materiale utilizzato a questo scopo è il silicio amorfo, buon isolante, che tramite il passaggio di alte correnti si trasforma in silicio cristallino, ottimo conduttore.
- Meccanismo di programmazione a PROM. Una ROM a fusibili può essere programmata sul campo dall'utente finale. L'interruzione del collegamento equivale alla programmazione di un '1' all'interno della cella corrispondente (la data line rimane fissa a livello alto).

Se il collegamento rimane, quando la word line si attiva il transistor si accende e porta a livello basso la data line. Le ROM di questo tipo possono essere programmate una sola volta.



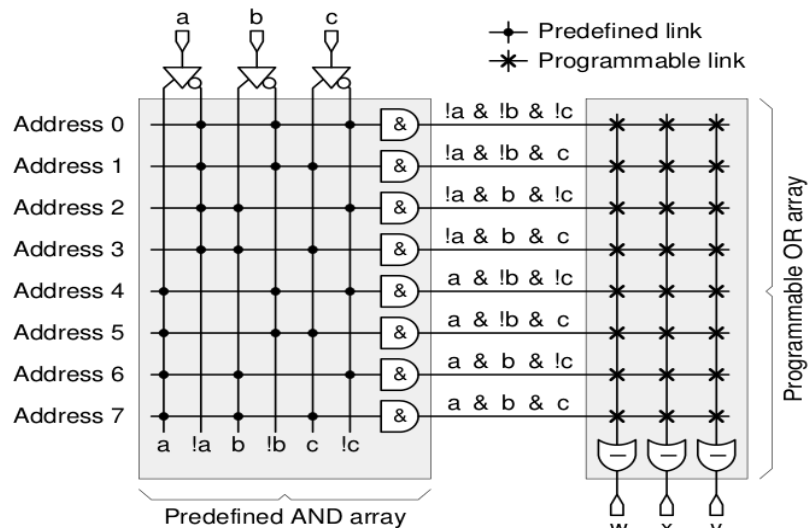
Programmazione PROM a fusibili

- Meccanismo a transistor MOS a gate flottante (FLOTOX). Essi forniscono un meccanismo di programmazione reversibile. L'accumulo di elettroni nel gate flottante provoca un aumento della tensione di soglia (relativa al gate di controllo) del transistor. Un accumulo sufficientemente alto fa sì che il transistor rimanga spento per tutte le tensioni operative del gate di controllo. La programmazione può essere cancellata eliminando gli elettroni dal gate flottante (ad esempio con radiazione UV).
- Meccanismo a cella con transistor a gate flottante. Le celle di questo tipo possono essere programmate elettricamente in modo individuale (portando ad una tensione elevata la riga e la colonna corrispondenti). Anche questo metodo è reversibile, infatti è possibile cancellare la programmazione esponendo le componenti a radiazione UV.
- Metodo ad E²PROM, con programmazione e cancellazione elettrica. L'aggiunta di un transistor MOS normale in serie al transistor a gate flottante permette la cancellazione individuale per via elettrica delle celle. Lo svantaggio principale è che una cella occupa un'area quasi doppia rispetto a quella di una EPROM, per cui si ottengono componenti meno densi. Una variante più densa della E²PROM è la memoria FLASH, in cui le celle sono cancellabili a banchi. Il metodo di programmazione a memorie FLASH è il metodo utilizzato nei CPLD più recenti.



I metodi di programmazione sopra indicati vengono utilizzati per programmare la matrice di contatti che va a formare la matrice delle connessioni. Per realizzare la funzione logica voluta il dispositivo ha a disposizione una serie di porte AND ed OR che vengono collegate in maniera tale da formare la funzione voluta. Per fare ciò esistono 3 configurazioni principali per collegare questi componenti:

- PLD ad architettura PROM: in questa configurazione la struttura tipica del PLD è basata su un piano AND predefinito e da un piano OR programmabile. Il piano AND fornisce tutti i mintermini, mentre attraverso la programmazione del piano OR è possibile selezionare i mintermini che si vogliono sommare per dare l'uscita.

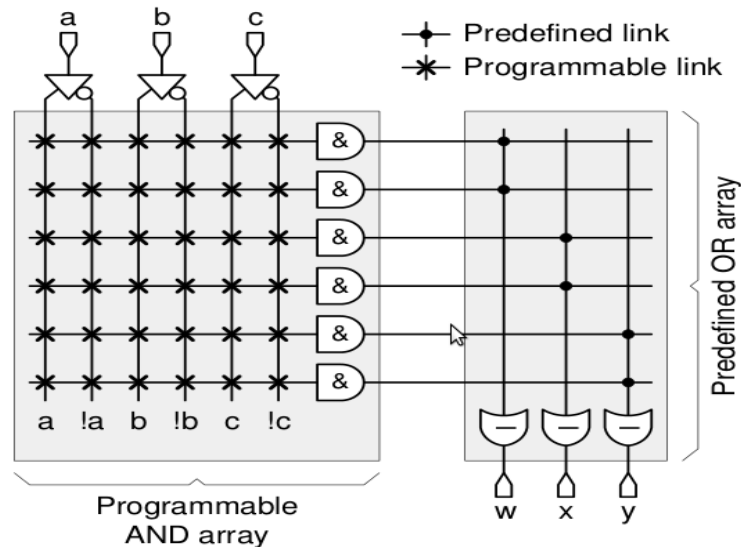


PLD con architettura PROM [9]

- PLD ad architettura PLA (Programmable Logic Array): in questa configurazione sono programmabili sia il piano AND che il piano OR. Se da una parte questa

configurazione permette una maggiore libertà di programmazione, dall'altra il fatto di dover attraversare 2 livelli di programmazione questa struttura rende il PLD molto più lento e quindi poco utilizzato.

- PLD ad architettura PAL (programmable Array Logic): nata per cercare di risolvere i problemi di velocità dell'architettura PLA pur mantenendo la programmabilità del piano AND. Questa configurazione permette una maggiore velocità, a costo però di aver meno flessibilità nella programmazione del piano OR.



PLD con architettura PAL [9]

Per la realizzazione di un CPLD viene utilizzata una matrice di PLD sopra descritti collegati da una rete di interconnessioni programmabili. I primi CPLD realizzati in questo modo furono un fallimento a causa dell'eccessivo consumo di potenza e alla scarsa velocità, entrambi causati dal fatto che si voleva mantenere la proprietà che ogni blocco potesse essere collegato a qualsiasi altro blocco. Per risolvere questo problema, Altera fu la prima a proporre una CPLD con connettività incompleta, realizzato in tecnologia CMOS e con celle di memoria EPROM.

3.2.5.2 Implementazione di un circuito

La progettazione di un circuito da implementare su un CPLD è relativamente semplice, grazie anche ai numerosi strumenti utilizzabili per la progettazione: è possibile utilizzare il linguaggio HDL per simulare il circuito e gli strumenti CAD per la sintesi del progetto. Per fare questo sono disponibili gratuitamente dei tool di sviluppo come l'ISE WEBPACK [10]

fornito gratuitamente da Xilinx, uno dei maggiori produttori di CPLD.

XILINX ISE WebPack

Con l'utilizzo di questo insieme di tool di sviluppo la progettazione viene suddivisa in 8 principali step:

1. Pianificazione del progetto;
2. Scrittura del codice VHDL o disegno dello schema circuitale;
3. Compilazione del file VHDL per ottenere il file RTL con cui procedere alla sintetizzazione;
4. Sintetizzazione del progetto fino a creare la netlist relativa;
5. Simulazione del progetto per verificarne il funzionamento;
6. Implementazione: questa fase include la traduzione da netlist a circuito vero e proprio, la mappatura del circuito e il suo posizionamento all'interno del dispositivo scelto;
7. Simulazione temporale: in questa fase è possibile simulare il circuito per individuare i tempi minimi di propagazione, necessari per individuare la massima frequenza di funzionamento;
8. Creazione del BIT file, che verrà utilizzato dal programmatore per programmare il dispositivo.

Una volta creato il file BIT, esistono due modi per caricarlo nella CPLD: la programmazione in circuit e la programmazione standard. La programmazione in circuit si avvale dello standard JTAG [11] e, tramite il pilotaggio dei segnali sui piedini dedicati presenti nella CPLD, permette di programmare e riprogrammare il dispositivo senza doverlo togliere dalla scheda. La programmazione standard invece si avvale dell'utilizzo di un programmatore esterno, con cui la CPLD viene programmato prima di essere montato nel circuito.

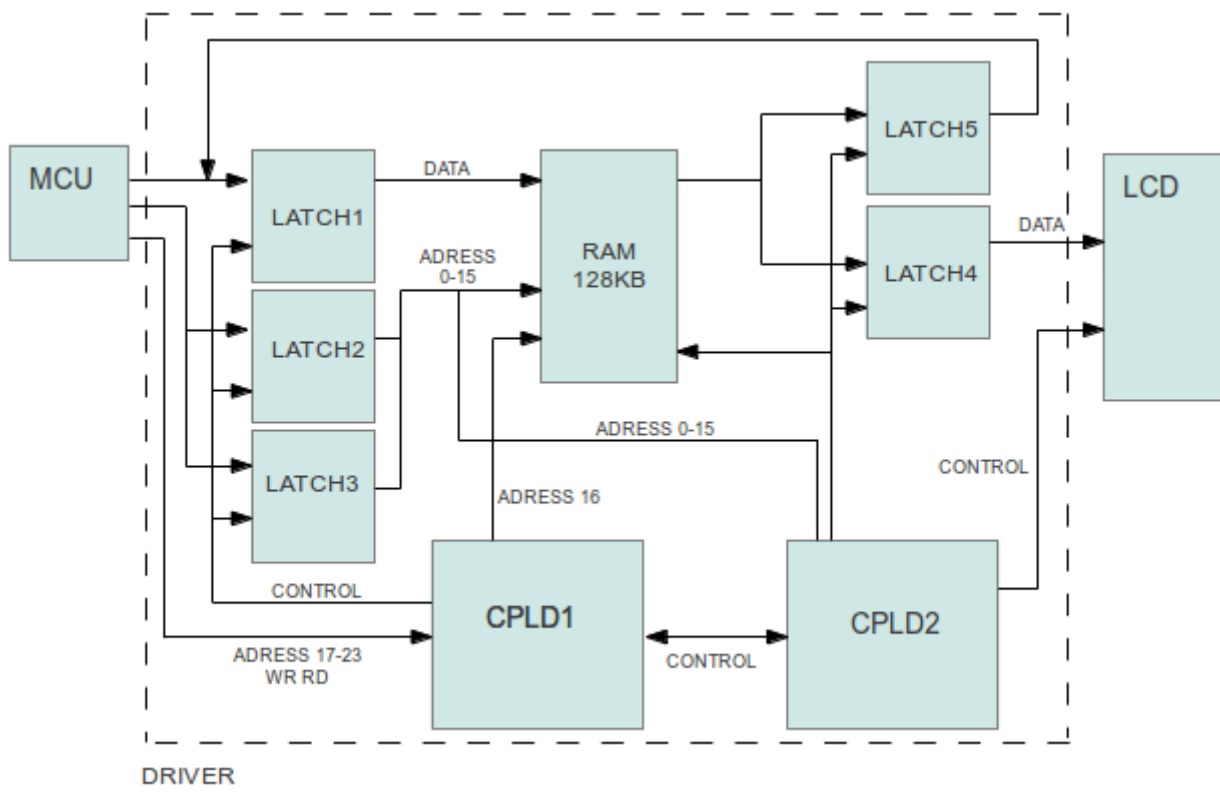
3.3 Principio di funzionamento della scheda

Il driver può essere inserito in un sistema di controllo in cui sono presenti anche altri dispositivi, e pilotato in modalità memory-mapped, con un range di indirizzi nei quali è possibile leggere o scrivere sul dispositivo. Per avere maggiore flessibilità di utilizzo, la parola che vado a scrivere sulla RAM può essere inserita in vari modi: posso dare direttamente in ingresso al driver parola ed indirizzo in cui voglio scrivere, oppure prima

scrivere la parola e poi gli indirizzi, o altre tipologie di immissione. Per realizzare ciò, sono stati inseriti in ingresso, sia sul bus indirizzi che sul bus dati, dei latch su cui, operando su diversi indirizzi, è possibile scrivere. Una volta che i valori di dati ed indirizzi sono stati scritti sui latch d'ingresso, agendo su un'ulteriore indirizzo è possibile segnalare al dispositivo l'avvenuta scrittura, e il dispositivo stesso provvederà ad eseguire le operazioni necessarie a salvare il dato in memoria.

E' stato possibile utilizzare la tecnica di memory-mapped perchè, dovendo indirizzare 128 Kbyte di RAM, sono necessari solo 17 dei 24 bit di indirizzo disponibili per indirizzare la memoria interna al dispositivo, mentre i restanti 7 più significativi identificano il componente su cui si vuole operare.

Per una più semplice comprensione della modalità di funzionamento del dispositivo, viene di seguito allegato uno schema a blocchi:



Si è deciso di utilizzare 2 CPLD separate perché in questo modo è possibile utilizzare un modello di CPLD che sia possibile montare direttamente in azienda. Una volta verificato il funzionamento del prototipo, si potrà utilizzare una CPLD unica nella realizzazione della scheda definitiva.

3.3.1 Interfaccia verso il microcontrollore

Per collegarsi al microcontrollore, si è deciso di ricorrere ad un connettore a 40 vie così suddiviso:

- 1-8 Bus dati: essendo la RAM organizzata ad 8 bit, si è dovuti dedicare 8 bit ai dati;
- 9-32 Bus indirizzi: collegati tramite latch, permettono l'utilizzazione di tutti e 24 i bit o di una porzione di essi in base al tipo di microcontrollore utilizzato;
- 33 pin di write: agendo su questo pin si va a scrivere nella locazione selezionata, che può essere di volta in volta una locazione di memoria vera e propria o un flag della scheda.
- 34 pin di read: permette la lettura di un certo indirizzo. Come sopra, questo indirizzo può rappresentare una locazione di memoria o un flag.
- 35 pin di controllo della retroilluminazione;
- 36-40 alimentazioni. La scheda è alimentata a 5 V. Viene prevista un'alimentazione separata tra i 5 V che alimentano la scheda e quelli che vanno ad alimentare la retroilluminazione dello schermo per limitare la corrente su ogni pin.

I pin dal n° 1 al n° 32 sono collegati alla scheda tramite dei latch con uscite di tipo tre-state, per evitare di incorrere in fenomeni di bus contention, come spiegato nella prossima sezione. L'interfaccia verso il microcontrollore è stata pensata con questa configurazione per fare in modo che la scheda possa essere vista e pilotata dal microcontrollore come una semplice periferica, in maniera tale da poter essere inserita in un bus comune a più dispositivi e quindi essere gestita in memory mapping.

3.3.1.1 Scrittura dell'immagine nella memoria

Per salvare un dato in memoria, il microcontrollore va a scrivere nei latch 1 i dati (pin 1-8), nei latch 2 e 3 (pin 9-24) gli indirizzi. Dato che per l'indirizzamento della RAM si utilizzano solamente 17 bit, i restanti bit di indirizzo vengono utilizzati dalla CPLD1 per la decodifica atta ad individuare a quale risorsa si vuole accedere. In questo modo, ponendo ad un valore adeguato i bit di indirizzo più significativi, è possibile decidere se scrivere su un latch alla volta, tutti e 3 assieme, o prima in quello dei dati e poi in quello degli indirizzi. In questo modo è possibile a MCU con architetture diverse utilizzare la scheda driver.

Per rendere possibile la scrittura di un latch alla volta, utilizzando i bit di indirizzo dal 18 al 24 sono stati individuati gli indirizzi dei vari “componenti”, interni alla scheda, che si vogliono andare a scrivere. Questi indirizzi sono riportati nella seguente tabella:

Indirizzo	Operazione
1111000	Scrittura su L1
1111001	Scrittura su L2
1111010	Scrittura su L3
1111011	Scrittura su L2 e L3
1111100	Scrittura su L1, L2 e L3
1111101	Scrittura flag PAGESEL
1111110	Scrittura flag LedOnOff
1111111	Scrittura flag abilitazione O4
1110000	Scrittura indirizzo per lettura (L2 e L3)
1110011	Scrittura flag Q

Il bit a destra è il bit meno significativo (bit 18 degli indirizzi) mentre quello a sinistra è quello più significativo (bit 24 degli indirizzi).

Sono stati riportati 2 indirizzi che vanno a scrivere entrambi sui latch 2 e 3 perché, essendo disponibile anche la possibilità di leggere un dato dalla RAM, andando a scrivere su uno o sull'altro indirizzo viene specificato alla scheda driver che tipo di operazione si vuole andare a fare.

Per andare a scrivere su un dato componente o flag, è necessario prima porre sul bus indirizzi l'indirizzo voluto e poi agire sul bit di WRITE.

Il flag Q riportato in tabella è un bit di stato utilizzato nella scheda driver per segnalare che nei latch di ingresso sono disponibili dei valori da utilizzare. Questi valori possono essere sia la parola da scrivere in memoria che l'indirizzo in cui scriverla, ma anche l'indirizzo che voglio andare a leggere. Più in generale, il flag Q indica che i valori sul bus dati o indirizzi non sono ancora stati utilizzati, e che quindi non è possibile eseguire inserimenti di nuovi dati fino a che essi non siano stati impiegati, pena la perdita dei dati stessi.

Viene di seguito inserito un esempio chiarificatore di un'operazione di scrittura, ipotizzando di utilizzare un microcontrollore a 8 bit, che deve quindi scrivere i valori di dati ed indirizzi 8

bit alla volta. In questo caso, potendo scrivere solamente 8 bit alla volta dovranno essere cortocircuitati i pin di ingresso in maniera tale da formare un unico bus a 8 bit, ognuno dei quali collegato a 3 pin nella maniera bit 1-9-17 cortocircuitati, bit 2-10-18 eccetera.

Queste sono le fasi in cui si divide l'inserimento del dato e dell'indirizzo:

- Viene posto sul bus il byte di dati;
- Viene posto sui bit più significativi del bus indirizzi l'indirizzo 1111000 che indica il primo latch, e si va ad agire sul bit di WRITE. In questo modo il valore viene latchato su L1;
- viene posto sul bus gli 8 bit meno significativi dell'indirizzo in cui si vuole scrivere il dato;
- Analogamente a prima, viene posto sui bit più significativi del bus indirizzi l'indirizzo 1111001 che indica il secondo latch, e si va ad agire sul bit di WRITE. In questo modo il valore viene latchato su L2;
- Nello stesso modo vengono salvati anche gli 8 bit più significativi;
- A questo punto viene scritto il flag Q, il quale indica che dati ed indirizzi possono essere utilizzati. In questo modo la CPLD 2 andrà ad agire sui vari valori di controllo per scrivere i dati inseriti nell'indirizzo specificato. Ciò che fanno le 2 CPLD verrà spiegato nella sezione dedicata alla programmazione.

In questo modo è possibile andare a scrivere una immagine intera nella RAM, immagine che verrà poi utilizzata dalla scheda driver per rinfrescare lo schermo LCD.

3.3.1.2 Lettura dell'immagine presente in memoria

La scheda driver permette inoltre la lettura dei dati memorizzati nella RAM. Per fare ciò viene usato un metodo simile all'operazione di scrittura già descritta. In questa fase la MCU scrive prima l'indirizzo che vuole andare a leggere utilizzando il metodo sopra descritto. A questo punto, la CPLD1 pone in uscita ai latch 2 e 3 gli indirizzi precedentemente memorizzati, e segnala alla CPLD2 che si è in fase di lettura. Questa predisporrà i segnali di controllo opportuni per andare a scrivere nel latch 5 il valore della locazione di memoria che si sta leggendo. Una volta scritto questo valore, lo segnala alla CPLD1 che, una volta ricevuto il segnale di read (RD) dalla MCU, mette in uscita tramite il bus dati i valori precedentemente

salvati nel latch 5.

Agendo sul bit di READ è possibile andare a leggere il valore salvato nel latch 5 oppure il valore del flag Q. Viene di seguito riportata la tabella di indirizzi su cui andare a leggere la parola in memoria o il flag Q:

Indirizzo	Operazione
1110001	Lettura di L5
1110010	Lettura flag Q

Anche in questa tabella il primo bit a sinistra è il più significativo (pin 24 del bus indirizzi) mentre quello a destra è il bit meno significativo.

In maniera simile a quanto spiegato nelle operazioni di scrittura, anche in questo caso è necessario prima porre in ingresso l'indirizzo della risorsa che si vuole leggere e poi agire sul pin di READ. Nella fase di scrittura è opportuno verificare il valore del flag Q, che fino a che viene mantenuto a livello logico alto indica che l'indirizzo da leggere non è stato utilizzato, e che quindi i dati salvati in L5 non corrispondono ai dati che si vuole andare a leggere.

3.3.2 Interfaccia verso il display

Per collegare l'uscita della scheda driver al connettore del display si è dovuto utilizzare un connettore a 24 vie, in maniera tale da poter connettere la scheda stessa al flat di ingresso del display. Per l'ordinamento dei pin si è seguita l'indicazione presente nel datasheet dello schermo, ordinamento riportato nella tabella presente nella prossima pagina :

Pin No.	Symbol	Level	Description
1	V5	---	Power supply pin for LCD driver voltage bias (VSS<V5<V4<V3<V2<VEE)
2	V2	---	Power supply pin for LCD driver voltage bias (VSS<V5<V4<V3<V2<VEE)
3	VEE	+26V	Power supply pin for LCD driver voltage
4	VDD	+5V	Logic system power supply pin
5	FLM	H/L	Frame signal pin
6	VSS	0V	Ground
7	CL1	H/L	Latch pulse input pin for display data
8	VSS	0V	Ground
9	M	H/L	AC signal input for LCD driving waveform
10	/D.OFF	L	Control input pin for output deselect level
11	CL2	H/L	Clock input pin for taking display data
12	V4	---	Power supply pin for lcd driver voltage bias (VSS<V5<V4<V3<V2<VEE)
13	V3	---	Power supply pin for lcd driver voltage bias (VSS<V5<V4<V3<V2<VEE)
14	D3	H/L	Data bit 4
15	D2	H/L	Data bit 3
16	D1	H/L	Data bit 2
17	D0	H/L	Data bit 1
18	NC	---	No connection
19	T/P(X2)	---	Touch panel signal (X2)
20	T/P(Y1)	---	Touch panel signal (Y1)
21	T/P(X1)	---	Touch panel signal (X1)
22	T/P(Y2)	---	Touch panel signal (Y2)
23	A	3.3V	White LED backlight anode
24	K	0V	White LED backlight cathode

Piedinatura flat LCD

Seguendo la tabella, sono stati collegati i pin di alimentazione alle tensioni opportune ricavate nella scheda tramite gli integrati MCP1825 e MC34063 per quanto riguarda le tensioni di alimentazione della retroilluminazione e della tensione di contrasto, mentre il pin 4 (VDD) è stato collegato direttamente all'alimentazione della scheda. I pin dati (pin 14-17) sono stati collegati al bus dati proveniente dalla memoria (nel modo spiegato nella sezione dello schematico) mentre i pin di controllo (FLM, M, CL1 e CL2) sono stati collegati al CPLD che genera i segnali di controllo come specificato nel grafico delle temporizzazioni.

Dato che il modulo LCD presenta un flat con passo di 0.5 mm si è reso necessaria la predisposizione di un doppio connettore a 24 vie, uno utilizzato per l'interfacciamento con il display e l'altro utilizzato come *test point*, come illustrato nella sezione dedicata allo schematico.

3.3.2.1 Visualizzazione sullo schermo LCD dell'immagine in memoria

Seguendo le specifiche di progetto, si è dovuto provvedere ad una frequenza di refresh pari a 50 Hz, onde evitare fastidiosi sfarfallii o immagini poco nitide. Per fare ciò, tramite la programmazione della CPLD2, che si occupa della generazione dei segnali di controllo necessari al display, sono stati generati i segnali opportuni. Basandosi sull'andamento dei

segnali riportato nel grafico delle temporizzazioni si è andati a generare i segnali FLM ed M tali in maniera tale da mantenere la frequenza di refresh voluta, cioè il segnale M con frequenza di 25 Hz e il segnale FLM normalmente a livello logico basso con una transizione a livello logico alto ad ogni nuovo frame da scrivere nel display, cioè un segnale rettangolare con frequenza di 50 Hz e permanenza a livello logico alto pari alla larghezza di uno dei segnali di clock (CL1 e CL2).

L'operazione di scrittura di un'immagine sullo schermo non causa nessuna difficoltà particolare, dovendo solamente seguire le indicazioni specificate dal produttore.

3.4 Programmazione della CPLD

Per gestire la scheda di controllo si è dovuto implementare un circuito con cui programmare le CPLD in maniera tale da generare tutti quei segnali necessari al funzionamento della scheda e al pilotaggio del display.

3.4.1 CPLD1

La CPLD1 si occupa principalmente della decodifica degli indirizzi per quanto riguarda la tecnica memory mapping e della generazione, conseguentemente, dei segnali opportuni necessari al corretto funzionamento della scheda. Per questo motivo, a differenza della CPLD2, questo dispositivo non ha bisogno di un ingresso di clock che causi l'aggiornamento dei segnali, ma si basa sulle variazioni dei segnali che riceve in ingresso per generare gli aggiornamenti.

Per operare la decodifica degli indirizzi il dispositivo si basa solamente sui 7 bit più significativi dei 24 bit in ingresso, dai quali ricava i segnali da generare. I principali segnali che interessano la CPLD1 sono:

- WR: segnale proveniente dall'ingresso MCU, va a segnalare che si vuole andare a leggere nella locazione di memoria specificata;
- RD: come sopra, specifica la locazione di memoria che si vuole andare a leggere;
- O1, O2, O3: uscite che abilitano le uscite dei relativi latch in base alla codifica degli indirizzi e dello stato del flag Z_ctrl proveniente dalla seconda CPLD;
- Q: flag che indica la presenza di nuovi dati nei latch di ingresso, che devono essere letti. Questo flag viene portato a livello logico alto scrivendo in una determinata

locazione di memoria, come specificato nella sezione relativa al principio di funzionamento della scheda. Una volta che il flag Q viene settato a livello logico alto, la CPLD2 si adopererà per andare a leggere i dati dai latch e, una volta letti, a resettare il flag. Questo flag viene letto dal microcontrollore per vedere se può andare a scrivere un nuovo dato nei latch di ingresso.

- R: segnale proveniente da CPLD2, usato per resettare il flag Q;
- RWS (Read Write Select): flag utilizzato per segnalare se si è in una fase di lettura o di scrittura della memoria. Per settare il valore di questo flag la CPLD1 si basa sui 7 bit più significativi dell'indirizzo posto in ingresso.

Le funzioni implementate da questa CPLD sono:

- Produzione dei corretti segnali ad ogni scrittura di nuovi indirizzi: questa funzione deve essere agganciata al segnale di WRITE della scheda, infatti ad ogni comando di scrittura la CPLD deve andare a leggere gli indirizzi ed in base al loro valore produrre quei segnali necessari all'espletamento della funzione voluta. Per fare ciò ci si basa sulla tabella degli indirizzi riportata nella sezione di lettura/scrittura di un'immagine. In sintesi, quando si va a scrivere un latch la scheda si occupa della generazione del segnale di latch enable per il latch voluto, se vado a scrivere il flag Q la CPLD aggiorna lo stato del flag e setta, in base al fatto che sia in lettura o scrittura, il flag RWS. Questo flag è portato ad 1 se sono in lettura, a 0 se sono in scrittura.
- Controllo delle uscite dei latch di ingresso: per fare ciò, la CPLD legge il segnale Z_ctrl proveniente dalla CPLD2 che indica se le uscite possono essere abilitate o devono essere mantenute nello stato alta impedenza.
- Controllo del valore del flag Q: questo flag viene scritto dal microcontrollore per segnalare che ha completato la scrittura dei dati e degli indirizzi nei latch di ingresso. Una volta che il flag Q è stato portato a livello logico alto, la CPLD2 genera gli opportuni segnali per l'utilizzo dei dati. Una volta che ha completato l'operazione, il flag Q viene resettato per segnalare alla MCU che può andare a riscrivere i dati nei latch di ingresso.

Quando si va a leggere o scrivere un flag, esso è riportato nel primo pin dati. In questo modo anche i flag vengono visti a tutti gli effetti come delle locazioni di memoria che possono essere scritte o lette.

Per permettere la scrittura dei latch ad ogni comando di scrittura viene letto l'indirizzo, ed in base a questo vengono generati i segnali di controllo opportuni. L'operazione di scrittura viene implementata come riportato nel seguente spezzone di codice::

```
process (WR,R)
begin

if (WR'event) and (WR='1') and (R='0') then
  case ADDRESS is

    when "1111000" =>
      L1<='1';      --attivo solo L1
      L2<='0';
      L3<='0';

    when "1111001" => --attivo solo L2
      L1<='0';
      L2<='1';
      L3<='0';
      .
      .
      .
    end case;

end if;

if (R='1') then
  -- resetto il flag Q

  Q<='0';
end if;
end process;
```

Si è dovuto inserire anche il reset del flag Q nel processo delle scritture per avere nello stesso processo la gestione del valore del flag (SET del flag tramite indirizzo e comando di WRITE, RESET del flag tramite il segnale R).

In fase di lettura, è possibile andare a leggere 2 entità: il flag Q o il dato salvato nel latch 5, dato presente nell'indirizzo precedentemente scritto. Dato che nella scheda il bus dati è unico, si è dovuto porre particolare attenzione per quanto riguarda il coordinamento dei segnali in fase di lettura, perché essendo presente nello stesso bus i segnali provenienti da 3 componenti (la CPLD1 per quanto riguarda il flag Q, il latch 5 e la MCU per quanto riguarda i dati) è facile incorrere in situazioni di bus contention. Per evitare ciò si è andati a pilotare tutti i componenti ogni qualvolta si va ad agire sul bit di READ. Viene riportato la parte di codice VHDL che si occupa della gestione degli opportuni segnali in maniera tale da evitare queste situazioni:


```

if (RD'event) and (RD='1') then
  if (ADDRESS="1110001") then --controllo di agire sulla scheda LCD
    DATA0<='Z';
    O3<='0';
    else if (ADDRESS="1110010") then --controllo indirizzo lettura
      --di Q
      DATA0<=Q;
      O3<='1';
      else O3<='1';
      DATA0<='Z';
      end if;
    end if;
end if;

```

Nel codice sopra riportato il segnale `DATA0` è collegato al primo pin dati, e viene mantenuto sempre in alta impedenza (`DATA0<='Z'`) a meno di quando si va a leggere all'indirizzo del flag `Q`.

Una volta implementato il circuito, si è andati a generare, tramite i tool di sintesi presenti nell'*ISE WebPack*, il file necessario alla programmazione della CPLD. In questa fase di sviluppo del progetto è possibile impostare alcuni parametri in maniera tale da indicare al compilatore quali prestazioni si vogliono prediligere, se è necessario prediligere la dimensione del circuito, per minimizzare il numero di macrocelle necessarie, o la velocità, che incrementa il numero di macrocelle utilizzate ma massimizza la frequenza massima di funzionamento, a scapito però della potenza utilizzata dal componente. Dato che nello sviluppo della scheda è trascurabile il consumo energetico dei componenti, non dovendo appoggiarsi a batterie o simili, si è deciso di prediligere la velocità del componente. A fronte di queste scelte è stato opportunamente settato il sintetizzatore. A questo punto il circuito descritto in linguaggio VHDL è stato sintetizzato, con i seguenti risultati per quanto riguarda le risorse utilizzate della CPLD:

Macrocelle	Pterms	Registri	Pin	Blocchi funzionali
14/72	30/360	13/72	26/34	38/216

Come è facilmente osservabile, la scelta del modello di CPLD è estremamente

sovradimensionata, ma è stata utilizzata ugualmente una CPLD da 72 macrocelle per simmetria con le specifiche della CPLD2, in quanto da un punto di vista dei costi conveniva l'acquisto di due componenti uguali con capacità maggiore rispetto all'acquisto di due componenti diversi.

Inoltre sempre dal report generato dal tool di sintesi si nota che tutte le macrocelle utilizzate sono in modalità “*High Performance Mode*”, a causa del fatto che nella sintesi vengono predilette le caratteristiche di velocità a scapito dei consumi.

Per verificare il corretto funzionamento del circuito implementato è stato utilizzato il simulatore presente nell'*ISE WebPack* fornito dal produttore della CPLD. Settando opportunamente i segnali di stimolo, il circuito descritto in VHDL è stato testato, ricavando i seguenti andamenti temporali dei segnali posti in uscita:



Andamento temporale dei segnali della CPLD1

Come è possibile osservare, ad ogni impulso del segnale di scrittura (WR), il circuito decodifica l'indirizzo a cui si sta scrivendo, ed attiva i segnali di scrittura opportuni per poter andare a scrivere nella risorsa selezionata, seguendo la tabella degli indirizzi sopra riportata. Essendo i segnali di stimolo settati dal tool di controllo, essi non seguono le specifiche di progetto. Ad esempio, nella simulazione vengono scritti nuovi valori nei latch di ingresso senza controllare lo stato del flag Q. Questo fattore comporterebbe un errore di funzionamento che causerebbe la perdita dei dati immagazzinati nei latch di ingresso.

Il segnale di scrittura (WR) e il valore degli indirizzi provengono dalla MCU, mentre il segnale di reset (R) è generato dalla CPLD2 una volta che sono stati utilizzati i dati

memorizzati nei latch di ingresso. Non potendo simulare il sistema formato da entrambe le CPLD, tutti i valori dei segnali esterni alla CPLD1 sono forzati dal tool di simulazione.

3.4.2 CPLD2

La CPLD2 si occupa principalmente della generazione dei segnali di controllo necessari per leggere e scrivere sulla RAM e di quelli necessari al pilotaggio del display.

Dato che per il pilotaggio dei vari dispositivi è necessaria una generazione ciclica di segnali, si è pensato di realizzare il controllo come una macchina a stati finiti, in cui l'aggiornamento di stato è sempre uguale (ad ogni aggiornamento si incrementa semplicemente lo stato) ed è causato da un segnale di clock esterno. In questo modo il funzionamento della CPLD2 può essere diviso in due blocchi principali, il primo che si occupa della gestione dei dati da leggere o scrivere in memoria e la seconda che si occupa del rinfresco dello schermo con i dati letti dalla memoria, generando quindi i segnali necessari. I principali segnali da andare a generare (oltre a quelli necessari al funzionamento del display illustrati nella sezione dedicata all'hardware) sono:

- **Z_ctrl**: questo segnale gestisce le uscite dei latch di ingresso (L1, L2 ed L3). Quando il segnale è a livello logico alto le uscite sono abilitate, mentre quando è a livello logico basso le uscite sono in alta impedenza. Deve essere opportunamente condizionato in maniera tale che le uscite dei latch siano abilitate solamente quando le uscite in comune alla CPLD2 sono in alta impedenza. Questo segnale è collegato agli output enable degli integrati di ingresso.
- **WRM e RDM** (write memory e read memory): questi segnali gestiscono la lettura e scrittura nella RAM, secondo le disposizioni riportate nel datasheet relativo.
- **Q e RWS**: segnali provenienti dalla CPLD1, indicano se ci sono dati che devono essere processati e se sono in fase di lettura o di scrittura sulla memoria, rispettivamente.
- **L1, L2, L3, L4**: questi segnali, collegati ai latch enable degli integrati di ingresso e di uscita, gestiscono la memorizzazione dei dati nei vari latch, a seconda della fase e della modalità di funzionamento.
- **LA, LB**: questi due segnali sono utilizzati per pilotare il buffer posto in uscita al latch L4. Essendo questo latch collegato ai pin dati dello schermo, ed essendo questi pin solamente 4, devo poter abilitare i 4 bit più significativi e i 4 meno significativi

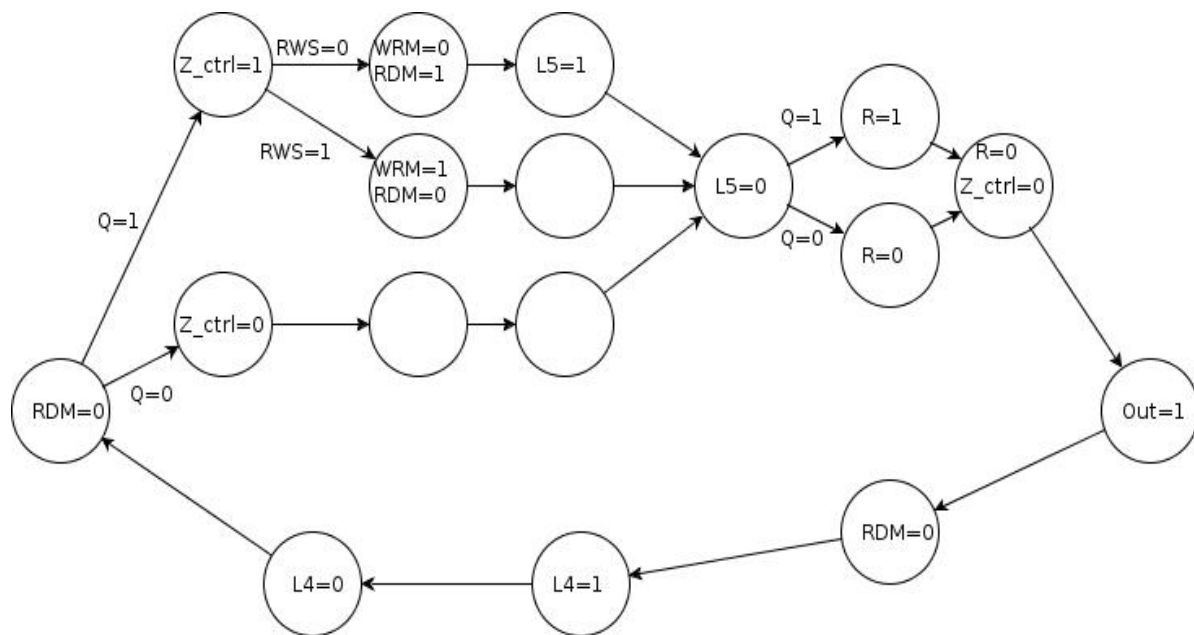
alternativamente, in maniera tale che con un byte memorizzato nel latch posso procedere con due operazioni di scrittura.

Vengono di seguito elencati i passi della macchina a stati finiti con specificato per ogni passo il segnale che si va ad aggiornare. L'aggiornamento dello stato è ciclico, ad ogni impulso di clock viene incrementato di una unità.

1. Controllo lo stato di Q. Se è a livello logico alto viene attivato il segnale Z_ctrl che pone in uscita ai latch i valori da utilizzare;
2. Se il flag Q era a livello logico alto, si va a controllare il valore del flag RWS (Read Write Select): se è alto significa che si vuole andare a leggere nella RAM, per cui viene portato a livello logico basso il bit RDM che mette in lettura la RAM, altrimenti si è in fase di scrittura nella RAM per cui verrà ad essere portato a livello logico basso il bit WRM. Se il flag Q era a 0 non è necessario aggiornare nessun segnale nei successivi 3 passi, perciò viene incrementato lo stato lasciando invariati i segnali;
3. Se si sta leggendo dalla memoria (RWS a 1), il valore letto viene memorizzato in L5 (L5 a livello logico alto) e reso disponibile per essere visualizzato in uscita;
4. Viene riportato a livello logico basso il bit L5. In questo modo il valore letto dalla memoria rimane memorizzato in L5;
5. Una volta utilizzati i dati presenti nel latch di ingresso, viene resettato il flag Q per segnalare che si può procedere ad una nuova scrittura nei latch. Viene anche disabilitato il flag Z_ctrl che pone in alta impedenza le uscite dei latch di ingresso;
6. Viene controllato il contatore di colonna: se è necessario incrementare la riga, il segnale CL1 viene portato a livello logico alto. Vengono abilitate le uscite indirizzi della CPLD. In questo modo si seleziona la locazione di memoria che si vuole andare a leggere per rinfrescare l'LCD;
7. Viene dato il comando di READ alla RAM. In questo modo vengono posti in uscita i valori desiderati, da utilizzare per il rinfresco dello schermo;
8. Viene attivato il segnale L4, che salva nel latch di uscita i valori letti.
9. Viene disabilitato il segnale L4. In questo modo i valori letti rimangono memorizzati nel latch.
10. Viene disabilitato il comando di lettura della RAM. A questo punto è possibile ricominciare un nuovo ciclo di scrittura.

Come accennato sopra, si può notare che nei primi 5 passi la CPLD gestisce i segnali necessari alla scrittura/lettura nella RAM, mentre nei secondi 4 si dedica ai segnali necessari al rinfresco del display.

Nello schema sotto riportato viene rappresentata la macchina a stati finiti realizzata nella CPLD2 e sopra descritta.



FSM realizzata nella CPLD2.

Per la scrittura dei dati sullo schermo LCD viene utilizzato il byte di dati memorizzato nel latch L4. Dovendo scrivere solamente 4 bit alla volta, viene utilizzato un buffer per trasferire sequenzialmente in uscita prima i 4 bit meno significativi e poi i 4 più significativi. Per la corretta acquisizione da parte del display, il segnale CL2 compie, nell'arco dei 10 passi della macchina a stati finiti, 2 fronti di discesa, uno quando sono abilitati i 4 bit meno significativi (attivo LA e disattivo LB) e il secondo per i 4 più significativi (disattivo LA e attivo LB). In questo modo in un ciclo completo della macchina a stati vengono scritti 8 bit, con 2 periodi completi di CL2. Per realizzare la macchina a stati finiti in maniera tale da poter essere implementata nella CPLD è stato usato un processo agganciato al segnale di clock, in cui in ogni stato vengono generati i segnali descritti sopra. Per la gestione degli ingressi si è utilizzato una variabile booleana di supporto, che viene posta nello stato TRUE quando si vogliono abilitare gli indirizzi e nello stato FALSE quando vengono posti in alta impedenza. Viene riportato uno stralcio del codice VHDL utilizzato per implementare la macchina a stati finiti.

```

process(clock)
begin

```

```

if (clock'event) and (clock='1') then
  case (stato) is
  when "0000" =>
    stato:="0001";
    L4<='0';
    RDM<='1';
    F2:=false;
    if (Q='1') then
      Z_ctrl<='1';
    end if;
    LA<='1';
    LB<='0';
    CL2<='1';
  when "0001" =>
    stato:="0010";
    if (Q='1') then
      if (RWS='1') then
        WRM<='1';
        RDM<='0';
      else WRM<='0';
        RDM<='1';
      end if;
    else RDM<='1';
      WRM<='1';
    end if;
    LA<='1';
    LB<='0';
    CL2<='1';
    .
    .
    .
  end case;
end if;

--uscita 3 state
if (F2) then
  address<=d;
  A16<=pageflag;
  else address<="ZZZZZZZZZZZZZZZZZZ";
    A16<='Z';
  end if;
end process;

```

Il bit `pageflag` viene utilizzato per selezionare l'immagine che si vuole andare a visualizzare, venendo utilizzato come bit più significativo dell'indirizzo.

Per l'incremento dell'indirizzo da utilizzare si è implementata una seconda routine agganciata al segnale `RDM` che fa incrementare di un'unità il contatore ad ogni nuova lettura. Il contatore deve essere resettato una volta che è stata ultimata la lettura della memoria. Quando viene

resettato il contatore viene generato anche il segnale FLM necessario al pilotaggio del display. Per fare ciò viene utilizzato un flag interno alla CPLD2 che, quando viene posto a livello logico alto, causa la generazione di un impulso su FLM e la transizione da livello logico basso a livello logico alto (o viceversa) del segnale M. Viene riportato il codice VHDL rappresentante il contatore, con all'interno la gestione del segnale FLM.

```

if (d<9600) then -- controllo se ho letto tutte le  locazioni
                --necessarie
    d:=d+1;

    else d:="000000000000000000";
end if;
if (count<40) then
    count:=count +1;
    F1:=FALSE;
    FLM<='0';
    else count := "00000000";
        if (count1<240) then
            count1:=count1+1;
            else count1:="00000000";
                FLM<='1';          --segnale di nuovo frame
                F1:=TRUE;
            end if;
        end if;
end if;

```

Il primo contatore (contatore di colonna) è impostato in maniera tale da arrivare fino a 40 perché, essendo il contatore incrementato dal segnale di lettura in memoria, ad ogni lettura vengono letti 8 bit, cioè $8*40=320\text{ bit}$, dati sufficienti a rinfrescare un'intera riga del display.

Si è dovuta individuare la frequenza di clock necessaria per soddisfare le specifiche di progetto.

Per fare ciò si è visto che per mantenere una frequenza di refresh pari a 50 Hz bisogna andare a scrivere $(320*240)*50=3840000\text{ bit/s}$. Sapendo che la scrittura di 8 bit comporta un ciclo completo del programma, cioè 10 passi, e che la macchina a stati compie un passo ad ogni ciclo di clock, si è dovuto inserire come generatore di clock un oscillatore con frequenza

$$\left(\frac{3840000}{8}\right)*10=4800000\text{ Hz}$$

. Vedendo la reperibilità degli oscillatori nel mercato, si è

andati a scegliere un oscillatore a 5 Mhz. Per diminuire la frequenza di aggiornamento che con l'utilizzo di questo oscillatore avrebbe superato la frequenza richiesta sono stati aggiunti dei passi vuoti che, pur non facendo nulla, allungano il tempo di ciclo in maniera tale da

rientrare nei tempi richiesti.

Una volta che è stato implementato il circuito, si è andati anche in questo caso a sintetizzarlo tramite l'uso dell'apposito tool disponibile nell'*Ise WebPack*. In questo caso si è dovuti però andare ad individuare un compromesso tra velocità e occupazione di spazio del circuito, in maniera tale da soddisfare i requisiti di velocità di aggiornamento delle uscite cercando però di mantenere un'occupazione di memoria contenuta. Per questo motivo sono state provate varie configurazioni del sintetizzatore, fino ad individuare la più opportuna. In questo modo si è arrivati ad un utilizzo delle risorse riassunto nella seguente tabella:

Macrocelle	Pterms	Registri	Pin	Blocchi funzionali
59/72	270/360	52/72	34/34	99/216

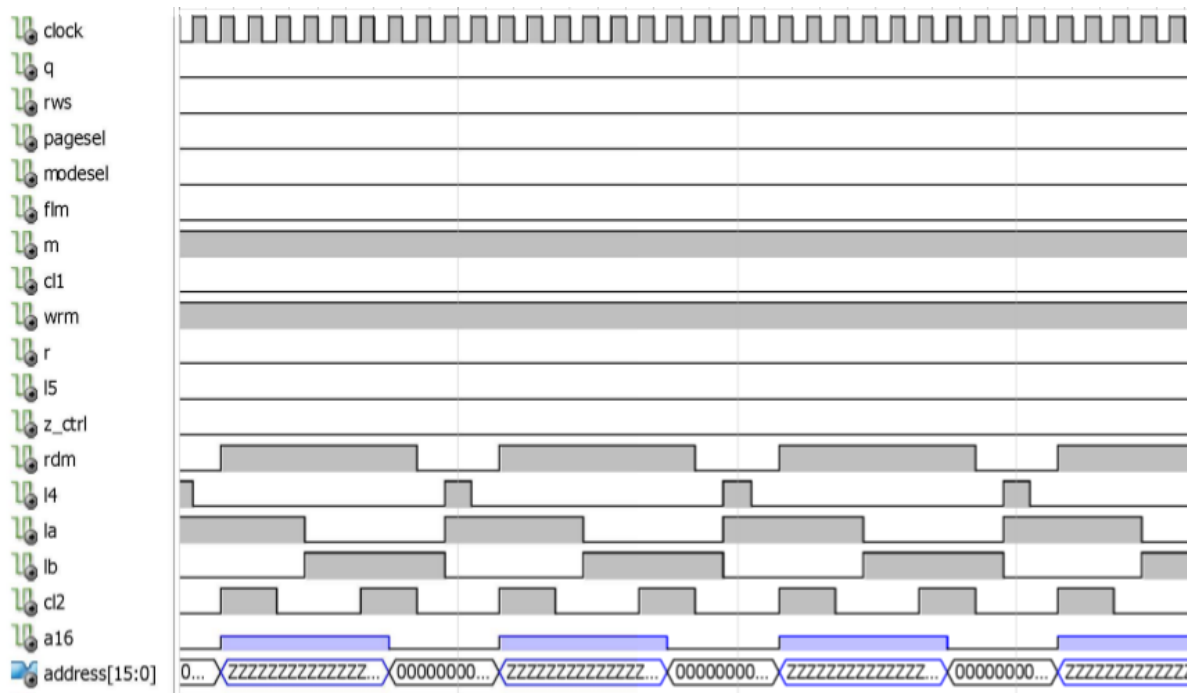
Anche in questo caso tutte le macrocelle utilizzate sono impostate in modalità alte prestazioni, per massimizzare la frequenza di funzionamento accettabile. Infatti, a differenza del circuito implementato nella CPLD1, il circuito implementato nella seconda CPLD deve soddisfare la specifica temporale riportata nelle specifiche progettuali, cioè deve generare gli opportuni segnali in maniera tale da permettere un refresh dello schermo con una frequenza di 50 Hz.

Per questo motivo il circuito, come specificato nella precedente sezione, deve accettare come sorgente di clock un segnale con frequenza di 5 MHz per permettere un tempo di refresh opportuno. Tramite il tool *Xilinx Timing Analyzer* presente nella *ISE Suite* si è verificato che il circuito implementato può funzionare ad una frequenza massima pari a 37 MHz, frequenza molto superiore a quella richiesta.

Come fatto per la CPLD1, una volta implementato il circuito tramite la descrizione in VHDL, questo è stato simulato con il tool di simulazione presente nell'*ISE WebPack* per verificarne il corretto funzionamento.

Anche in questo caso si sono dovuti forzare i segnali provenienti dall'esterno della scheda tramite la forzatura dei valori specificata nel tool di simulazione. In particolare si sono simulati i funzionamenti nelle tre modalità principali: la scrittura da parte della MCU di un nuovo dato in memoria, la lettura di un dato già presente e il funzionamento normale, dove la CPLD2 deve occuparsi solamente del refresh del display.

Viene ora riportato l'andamento temporale dei segnali controllati dalla CPLD2 quando non sono presenti dati da utilizzare nei latch di ingresso (modalità di funzionamento segnalata dal flag Q mantenuto a livello logico basso):



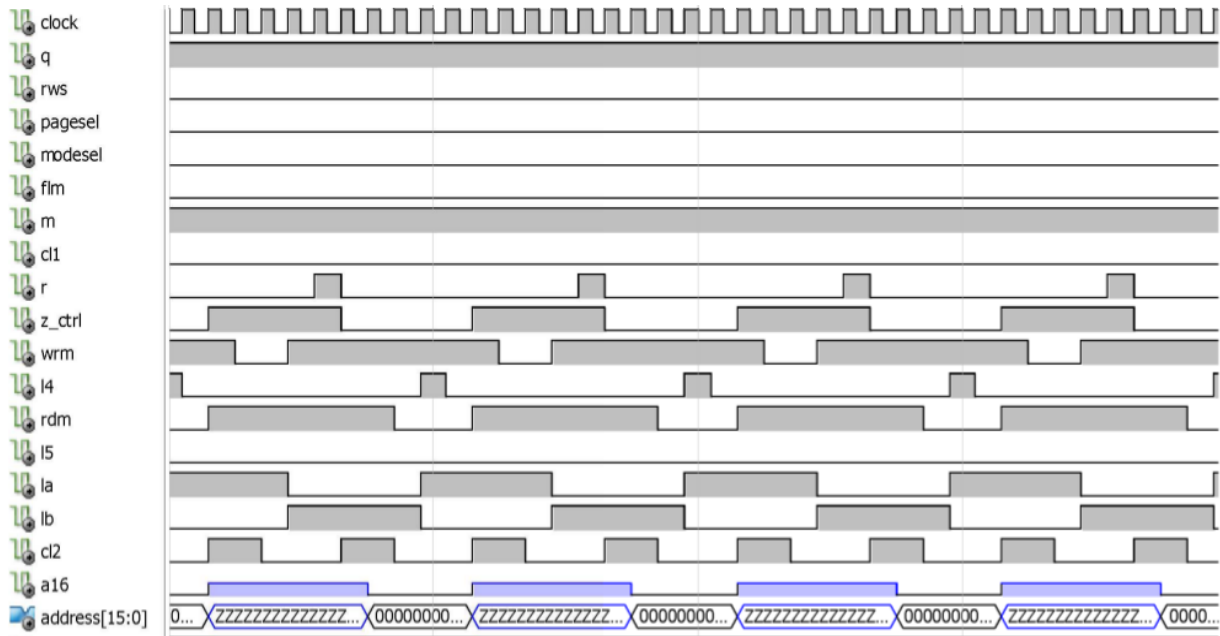
Andamento temporale dei segnali prodotti dalla CPLD2

Come illustrato precedentemente, per metà tempo di un ciclo completo la CPLD2 non fa nulla, limitandosi solo negli ultimi 4 passi ad operare per leggere i dati con cui refreshare la memoria. Infatti si può notare che viene prima posto in uscita l'indirizzo da utilizzare, poi settato il comando di lettura della RAM e infine agito su L4 in maniera tale da memorizzare i dati voluti nel latch di uscita. Dovendo scrivere 4 bit alla volta, tramite la gestione dei segnali LA ed LB che pilotano le uscite del buffer tre-state vengono posti in uscita 4 bit alla volta degli 8 memorizzati. Per ogni commutazione i bit vengono letti dallo schermo agendo sul segnale CL2.

Nella modalità di funzionamento con scrittura di nuovi dati o lettura di quelli già presenti in memoria varia il funzionamento nei primi 5 passi, mentre rimane invariata la fase di refresh dello schermo.

Viene di seguito riportato l'andamento temporale dei segnali in fase di scrittura di un nuovo dato in memoria da parte della MCU, la quale, una volta scritti i dati necessari nei latch di ingresso, segnala la presenza di nuovi dati andando a portare a livello logico alto il flag Q

controllato dalla prima CPLD. In questo modo la CPLD2 può generare gli opportuni segnali in base all'uso che deve fare dei dati memorizzati nei latch di ingresso, nel caso sia da scrivere un nuovo dato in memoria o sia da leggere una determinata locazione di memoria. Nel seguente diagramma temporale vengono riportati gli andamenti in una fase di scrittura di un nuovo dato, fase segnalata dal valore logico alto del flag Q e basso del flag RWS.

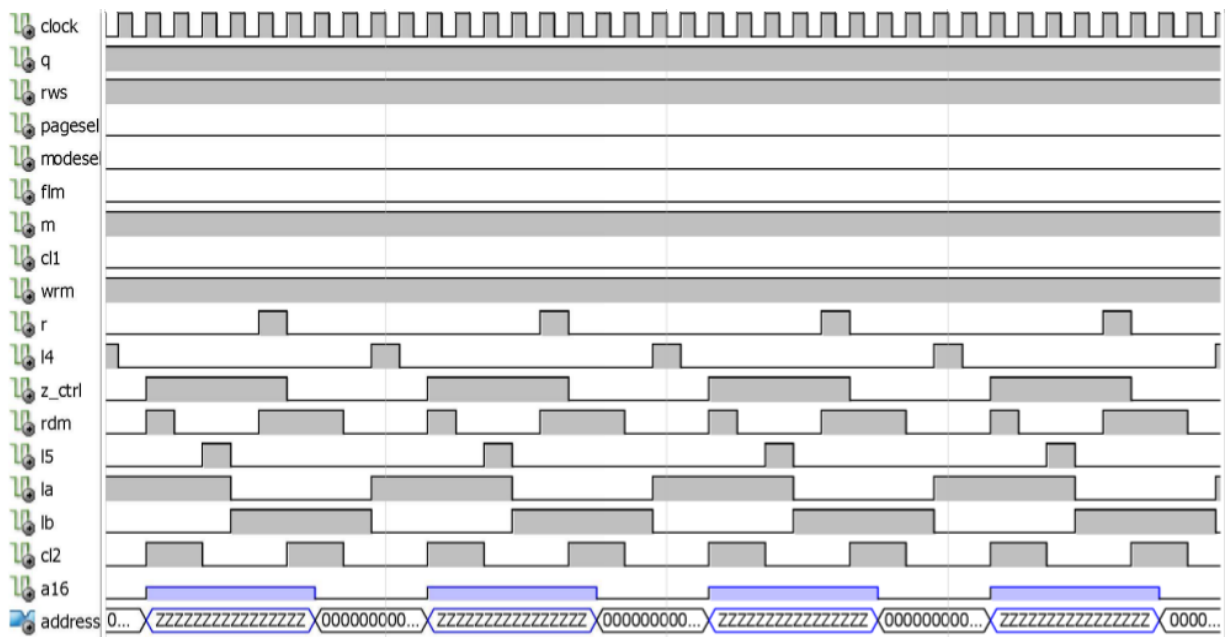


Andamento temporale dei segnali della CPLD2 in fase di scrittura di un nuovo dato

Come è possibile notare, per andare a scrivere è sufficiente porre a livello logico alto il segnale *Z_ctrl* che abilita le uscite dei latch L1, L2 e L3 su cui sono memorizzati il dato e l'indirizzo da scrivere. Si è dovuto fare attenzione affinché il segnale *Z_ctrl* non venga attivato prima di aver posto in alta impedenza l'uscita indirizzi della CPLD2, in maniera da evitare collisioni nel bus indirizzi.

Infine viene riportato l'andamento temporale dei segnali quando dalla MCU si vuole leggere un dato presente in memoria. Anche in questo caso una volta scritto l'opportuno indirizzo (dalla cui decodifica la CPLD1 ricava il valore che deve assumere il flag RWS) la MCU pone a livello logico alto il flag Q, che indica alla seconda CPLD la presenza nei latch di ingresso di nuovi valori da utilizzare.

Di seguito è riportato l'andamento temporale dei segnali quando nei latch di ingresso è presente un nuovo indirizzo da leggere. Questa modalità di funzionamento è segnalata alla CPLD2 dai segnali Q e RWS, entrambi a livello logico alto.



Andamento temporale dei segnali della CPLD2 in fase di lettura di un dato presente nella RAM

Analogamente a quanto fatto nella fase di scrittura, anche in questa modalità di funzionamento *Z_ctrl* deve essere pilotato in maniera tale da evitare collisioni nel bus indirizzi. Come si può osservare dallo schema temporale, questa modalità di funzionamento è composta da due letture uguali in memoria, la prima fatta utilizzando gli indirizzi disponibili nei lach di ingresso e salvando i dati in L5, mentre la seconda fatta utilizzando gli indirizzi del contatore interno e salvando i dati nel latch 4.

3.5 Lo schematico

Una volta individuato il principio di funzionamento della scheda, si è passati alla stesura dello schematico del circuito.

3.5.1 Individuazione blocchi funzionali

Una volta sviluppate le specifiche di progetto ed individuato il principio di funzionamento che deve avere la scheda, sono stati definiti 4 blocchi funzionali principali:

- Blocco di ingresso: in questo blocco funzionale possiamo inserire tutti quei componenti che si occupano dell'interfaccia verso il microcontrollore, che quindi permettono il corretto dialogo tra scheda driver e MCU. Questo blocco funzionale è costituito quindi dai latch che, oltre a rendere possibile la gestione della scheda tramite microcontroller diversi, scegliendo opportunamente la tecnologia costruttiva rendono possibile l'interfacciamento con MCU diverse anche per quanto riguarda le tensioni di funzionamento. Infatti scegliendo un integrato 5 V tolerant si ha la possibilità di controllare la scheda sia con dispositivi funzionanti a 3.3 V che con dispositivi a 5 V. I componenti di questo blocco, dovendo comunicare con l'interno della scheda, sono tutti alimentati a 3.3 V.
- Blocco alimentazioni: in questo blocco funzionale rientrano 3 componenti: due integrati MCP1825 che, ricevendo 5 V in ingresso, danno in uscita una tensione stabilizzata a 3.3 V e un integrato MC34063 che ricevendo in ingresso sempre 5 V, dà in uscita i 26 V necessari al pilotaggio del display. Sono stati inseriti 2 regolatori di tensione a 3.3 V perché, dato che la retroilluminazione del display può assorbire fino a 360 mA, si è preferito tenere separate le alimentazioni degli integrati della scheda dalla linea di alimentazione dei led della retroilluminazione. Per questo motivo sono presenti due ingressi separati per i 5 V che andranno sul primo integrato e quelli che andranno sul secondo. Per distinguere le due linee di alimentazione, si è ricorsi alla desinenza *LED* dopo l'indicazione della tensione (5V_LED e GND_LED nello schematico). I componenti di questa scheda, dovendo operare su regolazioni di tensioni, sono tutti alimentati a 5 V.
- Blocco di controllo: fanno parte di questo blocco tutti i componenti che non interagiscono direttamente con l'esterno della scheda, più precisamente quei componenti che si occupano della generazione/decodifica e della memorizzazione dei segnali (CPLD e RAM). Questo blocco è l'unico che contiene logiche programmabili (CPLD) e che quindi presenta la maggior complessità. Anche i componenti di questo blocco funzionale sono tutti alimentati ad una tensione di 3.3 V.
- Blocco di uscita: in questo blocco funzionale vengono inseriti i componenti che si occupano della trasmissione verso l'esterno dei segnali generati nella scheda. Fanno parte di questo blocco funzionale i latch di uscita (L4 ed L5) che, dovendo comunicare con componenti a 5 V come il display LCD, sono alimentati anche loro alla stessa tensione. I componenti di questo blocco funzionale si occupano quindi della

compatibilità tra scheda e componenti esterni, perché pur ricevendo in ingresso tensioni nel range 0-3.3 V, il loro valore in uscita soddisfa le specifiche dei componenti a 5 V.

3.5.2 Stesura dello schematico

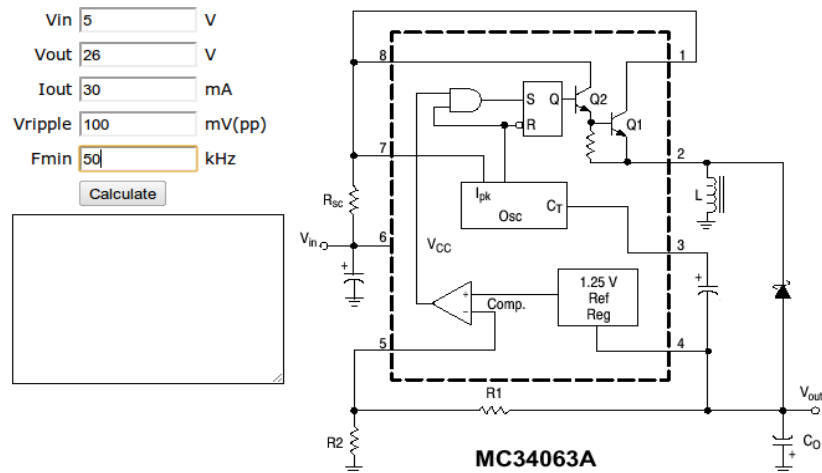
Una volta individuati i blocchi funzionali da cui deve essere composto lo schema circuitale della scheda, si è per prima cosa andati ad individuare il modello di componente che soddisfacesse le specifiche richieste e si sono individuate le configurazioni circuitali adatte. Vengono di seguito riportati gli integrati scelti e il dimensionamento dei componenti esterni.

- Latch: per la scelta dei latch si è dovuto tenere solamente conto della tolleranza a segnali in ingresso che superano la tensione di alimentazione. Per questo motivo si è scelto l'integrato 74HC373. Questo integrato presenta al suo interno 8 latch D binari le cui uscite sono collegate a dei buffer tre-state. Il funzionamento del componente è determinato dai segnali LE (latch enable) e OE (output enable): il primo controlla l'attivazione dei latch D, mentre il secondo controlla le uscite tre-state. Le uscite tre-state prevengono la possibilità di bus contention che potrebbero avvenire dato che il bus indirizzi e il bus dati è comune a più dispositivi. E' stato scelto l'integrato della famiglia HC per avere la tolleranza ai segnali in livello TTL in ingresso. Per il funzionamento di questo integrato non sono necessari componenti esterni, ad eccezione di un condensatore di bypass per assicurare la stabilità della tensione di alimentazione.
- Buffer tre state: dato che questo componente serve per interfacciare segnali a 3.3 V provenienti dalla scheda con lo schermo LCD che necessita di segnali a 5 V, si è dovuto individuare un integrato che pur essendo alimentato a 5 V riconoscesse correttamente i livelli di tensione dei segnali in ingresso. Per questo motivo si è scelto l'integrato 74HC244.

Questo integrato ha all'interno 8 buffer non invertenti con uscita tre-state. L'uscita tre-state è controllata da due pin, OE1 ed OE2, che controllano ciascuno la metà dei buffer presenti. Quando questi pin sono a livello logico basso il buffer riporta in uscita il valore che è presente in ingresso, mentre quando sono a livello logico alto l'uscita dei buffer è in alta impedenza. Anche questo integrato non necessita di altri componenti esterni, ad eccezione del solito filtro di bypass necessario a stabilizzare

l'alimentazione.

- RAM: seguendo le specifiche di progetto e la disponibilità di componenti già presenti in azienda, è stata scelta la SRAM CY7C1019DV, una ram 128K x 8 bit prodotta dalla Cypress Hill Semiconductor. Questo componente è una Ram statica ad alte prestazioni, presentando un tempo di accesso di soli 10 ns. Essendo disponibile con package SOJ, permette di essere montata a mano pur essendo un componente SMD. Questo integrato presenta 3 pin di controllo: chip enable (CE), output enable (OE) ed read enable (RE). Il primo è utilizzato per attivare la memoria, mentre i secondi due determinano se la memoria è in lettura o scrittura. Una spiegazione più dettagliata del funzionamento del componente è disponibile nel datasheet. Anche questo componente non necessita di componenti esterni a meno del solito condensatore sull'alimentazione.
- Step-Up converter: per ricavare la tensione di 26 V necessari per la corretta visualizzazione sullo schermo è stato usato l'integrato MC34063 prodotto dalla ON Semiconductor. Il suo funzionamento è già stato illustrato nella sezione dedicata all'hardware. Per il dimensionamento dei componenti esterni è stato utilizzato l'applicativo [12], che si basa sulle formule presenti nel datasheet per dimensionare i componenti necessari. Viene di seguito riportato un esempio dell'applicativo usato:



Applicativo utilizzato per il dimensionamento dei componenti

Utilizzando come tensione di ingresso i 5 V di alimentazione, tensione di uscita i 26 V necessari al contrasto, una massima variazione in ampiezza (V ripple) di 100 mV e una frequenza di apertura dello switch di 50 Hz, sono stati ricavati i seguenti valori:

- L: 180 μ H con massima corrente sopportata di 600 mA;

- R1: realizzata tramite la serie di un trimmer e una resistenza, per poter variare la tensione di uscita, deve valere complessivamente 20.8 K Ω . Per questo motivo è stato scelto un trimmer da 10 K Ω in serie ad una resistenza da 15 K Ω ;
- R2: necessaria a fornire la tensione di riferimento tramite il partitore resistivo con R1, viene utilizzata una resistenza da 1 K Ω ;
- Rsc: utilizzata per limitare la corrente che va a scorrere nell'induttanza, seguendo le specifiche del costruttore in questa modalità di funzionamento, viene utilizzata una resistenza da 1 Ω ;
- R3: resistenza utilizzata per limitare la corrente assorbita dalla connessione Darlington. Seguendo le indicazioni del manuale del produttore, è stata utilizzata una resistenza con valore standard di 180 Ω ;
- Co: essendo il condensatore utilizzato per la stabilizzazione della tensione di uscita, viene dimensionato basandosi sulle specifiche di stabilità che deve avere la tensione di uscita. Seguendo le specifiche del costruttore e l'applicativo sopra riportato, viene utilizzato un condensatore da 10 μ F elettrolitico;
- Ct: condensatore che determina la frequenza di apertura/chiusura dello switch interno, viene utilizzato un condensatore da 620 pF ceramico;

Per gli altri componenti, non dipendendo dalla modalità di utilizzo, sono stati utilizzati i valori standard indicati dal manuale. Per le resistenze si sono utilizzate tutte resistenze da ¼ di watt, ad eccezione della Rsc che, dovendo sopportare una corrente massima pari a 600 mA, deve essere in grado di dissipare una potenza di 0.36 W

Per questo motivo è stata scelta una resistenza da ½ watt.

Non avendo familiarità con il componente, per il dimensionamento sono stati impostati i valori in maniera tale da avere un certo margine di sicurezza. Ad esempio, dal datasheet del display è riportata una tolleranza in ingresso alla tensione di riferimento V_{ee} di 2V pp, mentre nel dimensionamento è stato impostato una massima variazione di 0.1 V, anche per evitare di incorrere in fastidiosi fenomeni di variazione del contrasto. Dato che anche in azienda era la prima volta che veniva utilizzato questo componente, è stato realizzato un prototipo per verificarne il funzionamento.

- Stabilizzatore di tensione a 3.3 V: necessario per ricavare la tensione di alimentazione

richiesta dai componenti della scheda e dalla retroilluminazione dello schermo, è stato scelto l'integrato MCP1825 prodotto dalla Microchip. A differenza del precedente, questo regolatore può dare in uscita una sola tensione stabilizzata, e non necessita di componenti esterni ad eccezione di un condensatore in ingresso e uno in uscita per mantenere stabili le tensioni di alimentazione. Sono stati utilizzati 2 integrati perché, dato che uno di questi può erogare fino a 500 mA di cui 360 vengono assorbiti dallo schermo, è preferibile mantenere 2 diverse linee di alimentazione a 3.3 V, una per la scheda e una per lo schermo.

- CPLD: per la scelta di questo componente si sono dovuti prendere in considerazione vari fattori. Prima di tutto la tensione di funzionamento: per poter avere buone prestazioni temporali, e dato che altri componenti necessitavano della stessa tensione, la scelta è ricaduta su un CPLD della serie XC9500XL della Xilinx. Questi dispositivi sono dei CPLD ad alte prestazioni alimentati a 3.3 V. Per scegliere il modello specifico di CPLD era necessario sapere il numero di macrocelle necessarie per implementare il circuito di controllo voluto. Per questo motivo si è dovuti prima implementare il firmware della CPLD in maniera tale da conoscerne l'occupazione di spazio. Per rendere possibile il montaggio delle CPLD in azienda, si è deciso di utilizzare 2 CPLD con package PCC44 al posto di una CPLD unica, che avrebbe determinato l'utilizzo di un package SMD difficilmente montabile in azienda. Alla luce di queste considerazioni, si è scelto di utilizzare due CPLD XC9572XL. Queste sono due dispositivi ad alte prestazioni con una disponibilità di 72 macrocelle all'interno, operano a 3.3 V e permettono di impostare ogni pin come ingresso o uscita. I pin di uscita possono essere definiti di tipo tre-state, fattore indispensabile per evitare episodi di bus contention data la presenza nello stesso bus di più componenti. Per la programmazione delle CPLD sono disponibili 2 metodi, come illustrato nella sezione dedicata all'hardware, in standard ed in circuit. Per la programmazione in circuit la CPLD presenta 4 pin dedicati, con cui è possibile programmare il dispositivo secondo lo standard JTAG.

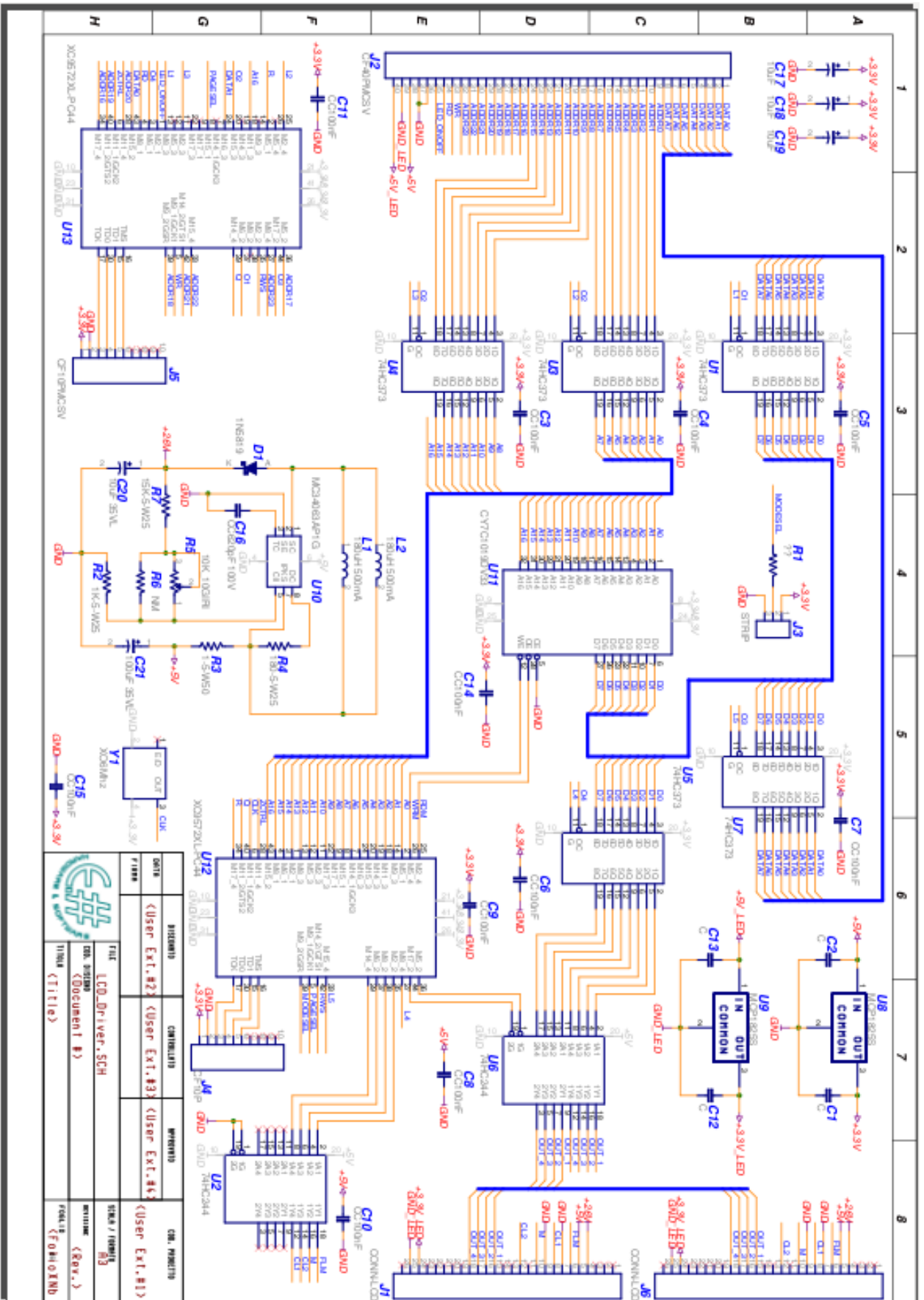
Per la corretta temporizzazione del funzionamento della CPLD, si è reso necessario l'inserimento di un oscillatore che fornisca al dispositivo il clock di funzionamento. Per questo motivo si è scelto un oscillatore SMD che fornisca un segnale di clock a 6 Mhz, come illustrato nella sezione relativa alla CPLD2. L'oscillatore non necessita di componenti esterni per il funzionamento e rende disponibile un segnale ad onda quadra con ampiezza pari a quella dell'alimentazione e frequenza fissa.

Una volta individuati i componenti da utilizzare, si è andati a cominciare la stesura dello schematico vero e proprio.

Dopo una serie di aggiornamenti e di correzioni suggerite dal tutor aziendale, si è arrivati a definire lo schematico definitivo del circuito.

Si noti che è stato scelto di inserire un doppio connettore in uscita. Questo è stato determinato dal fatto che, utilizzando un connettore per il flat LCD a tecnologia di montaggio superficiale, si avrebbe avuto il problema dell'accesso ai pin nella fase di collaudo. Per questo motivo si è deciso di inserire un secondo connettore da utilizzare come *test point* da utilizzare per rendere possibile l'accesso ai pin di uscita da parte del puntale di uno strumento.

Viene di seguito riportato lo schematico completo della scheda.



3.5.3 Individuazione errori

Una volta definito lo schematico completo, si è proceduto con una serie di controlli per individuare eventuali errori commessi durante la stesura dello schematico. Si è per prima cosa andati a controllare i bus dati ed indirizzi, che essendo in comune a più componenti potevano più facilmente presentare connessioni uscita-uscita che avrebbero determinato situazioni di conflitto di tensioni, potendo causare cortocircuiti e altri episodi di bus contention. Partendo dal bus dati, questo è collegato a 4 componenti: la RAM parte dati, i latch L1 parte uscite e L4, L5 parte ingressi. In questo caso si è solamente controllato che nella gestione dei componenti quando la memoria poneva in uscita dei valori le uscite di L1 fossero sempre in alta impedenza per evitare collisioni.

Per quanto riguarda il bus indirizzi si è dovuti invece fare attenzione al pilotaggio di CPLD e latch indirizzi (L2 ed L3). In questo caso, essendo possibile mettere in alta impedenza sia le uscite del latch che quelle della CPLD, si è dovuto fare attenzione alla corretta gestione degli opportuni segnali di controllo. In questi controlli si è reso utile il programma di controllo errori (ERC) disponibile con il programma utilizzato per lo schematico. In questo modo, settando opportunamente gli avvisi da dare nei vari tipi di connessioni, si è potuto avere anche una conferma da parte del programma della totale assenza di errori.

Una volta controllata l'assenza di errori nello schema elettrico della scheda, si è passati alla fase realizzativa.

Capitolo 4: Realizzazione e testing del prototipo

In questo capitolo si affronterà la parte più pratica del progetto, vale a dire la fase di disegno della scheda driver e delle successive fasi riguardanti la realizzazione del prototipo.

4.1 Realizzazione prototipo

Una volta prodotto lo schematico definitivo, si è passati alla fase realizzativa, che ha come obiettivo la realizzazione di un prototipo funzionante della scheda driver.

Prima di andare a realizzare il prototipo completo è stato prodotto un prototipo della parte riguardante la produzione della tensione di riferimento necessaria per il contrasto dell'LCD (+26 V). Si è dovuto realizzare questo prototipo per sincerarsi del corretto funzionamento dell'integrato MC34063, dato che nel progetto viene utilizzato non come alimentatore ma per fornire una tensione di riferimento, e a causa del fatto che nell'azienda è la prima volta che viene utilizzato.

4.1.1 Realizzazione prototipo alimentazione

Per verificare il funzionamento dell'integrato MC34063 si è proceduto con la realizzazione su basetta millefori dell'alimentatore a 26 V. Utilizzando il tool di calcolo precedentemente riportato, sono stati dimensionati i vari componenti. Uno dei motivi della realizzazione del prototipo è verificare il corretto dimensionamento dell'induttanza, di cui si conosce il valore ma non la massima corrente che deve sopportare. Per questo motivo si sono presi 3 tipi di induttori, da 150, 180 e 220 uH, con varie correnti massime sopportate. Per prima prova si è utilizzata un'induttanza da 150 uH e massima corrente supportata 280 mA. Una volta montato il circuito, si è proceduto con un primo controllo delle connessioni tramite l'uso di un tester che ne verificasse il collegamento elettrico. Una volta verificato il corretto montaggio, se ne è verificato il funzionamento ricavando i seguenti valori:

- Circuito a vuoto:

Vout : 26 V con ripple trascurabile.

Corrente assorbita dal circuito: <1 mA.

- Carico resistivo di 3.3 K Ω :

Vout: 26 V con ripple di 50 mV pp.

Corrente assorbita dal circuito 48 mA.

$$\text{Rendimento: } \frac{P_{RI}}{P_{assorbita}} = \frac{26 * I_l}{5 * i_{assorbita}} = 0.85$$

- Carico resistivo di 1650 Ω

Vout: 18 V con ripple di 150 mV pp.

Notando che la tensione non rispettava più le specifiche di progettazione, si è andati a sostituire l'induttanza con un'altra di uguale valore ma con corrente massima sopportabile di 550 mA, seguendo i consigli di progettazione reperiti su vari forum online [14] che trattano l'utilizzo di circuiti basati su questo componente. Con questa nuova configurazione si sono ottenuti i seguenti valori:

- Circuito a vuoto:

Vout 26 V con ripple trascurabile.

- Carico resistivo da 3.3 K Ω

Vout 26 V con ripple trascurabile;

Corrente assorbita dal circuito: 44 mA.

$$\text{Rendimento } \frac{P_{RI}}{P_{assorbita}} = \frac{26 * I_l}{5 * i_{assorbita}} = 0.92$$

- Carico resistivo da 1650 Ω

Vout 26 V con ripple di 40mV

Corrente assorbita dal circuito: 105 mA

$$\text{Rendimento } \frac{P_{RI}}{P_{assorbita}} = \frac{26 * I_l}{5 * i_{assorbita}} = 0.80$$

Si è utilizzato un carico minimo di 1650 Ω dato che le specifiche indicano che il massimo assorbimento su VEE del display è di 10 mA, e si è quindi voluto sovradimensionare il circuito del 50% onde evitare di incorrere in problemi.

Una volta verificato il funzionamento del convertitore, si è proceduto con la realizzazione del master dell'intera scheda.

4.1.2 Il master

Per questa fase del tirocinio è stato fondamentale l'apporto del componente dell'azienda che si occupa della realizzazione dei master.

4.1.2.1 Estrazione della netlist e lista componenti

Una volta completato e controllato lo schematico, si è passati all'estrazione della netlist. Trattasi di un file di testo in cui viene riportata ogni net del circuito e della lista dei componenti ad esso collegati. Una volta ottenuta la netlist si è dovuta modificarla in maniera tale da renderla compatibile con il programma *Altium*. Per fare ciò è stato utilizzato uno script in linguaggio C che va a prendere la netlist prodotta e la va a reimpaginare eliminando gli spazi non necessari tra le righe e seguendo la formattazione necessaria per la corretta comprensione da parte del programma utilizzato per il master. Una volta modificata opportunamente, la netlist è stata acquisita dal programma Altium che ha provveduto a dare una rappresentazione grafica del circuito, in cui le connessioni erano rappresentate da una serie di “elastici” che connettono i pin voluti di ogni componente.

Una volta creata la netlist è stata prodotta anche la lista componenti necessaria per indicare i componenti da usare per la realizzazione della scheda. Con questa lista si è andati a recuperare i componenti necessari, seguendo per la scelta le direttive riportate nel capitolo relativo allo schematico. Una parte dei componenti è stata recuperata direttamente dal magazzino dell'azienda mentre i restanti componenti sono stati ordinati presso i fornitori.

4.1.2.2 Individuazione footprint

Una volta terminata la stesura dello schematico, si è andati ad assegnare ad ogni componente il rispettivo footprint, se presente, utilizzato nel programma impiegato per la stesura del PCB. Dovendo utilizzare vari componenti SMD, tecnologia di montaggio poco utilizzata in azienda,

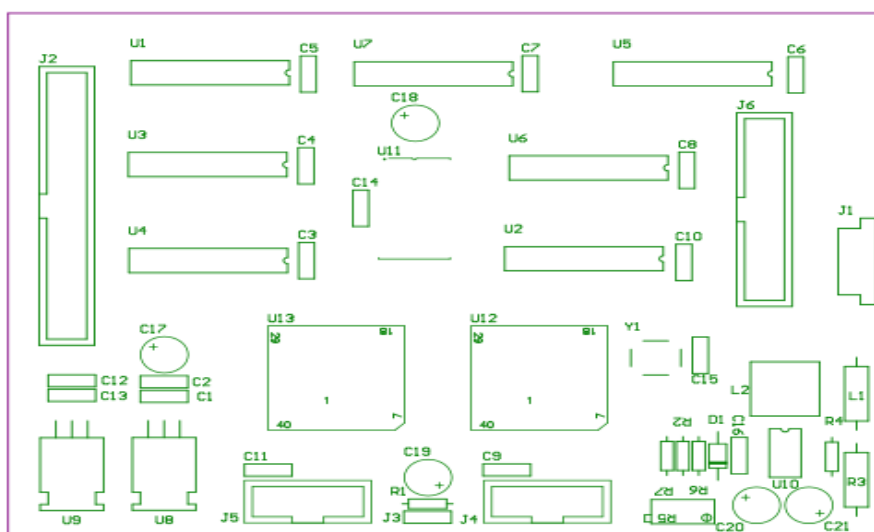
si è dovuto andare a disegnare come nuovi i footprint di questi componenti, in particolare quelli della RAM e dell'oscillatore. Per fare ciò ci si è basati sulle dimensioni riportate nei vari manuali.

Per il montaggio della CPLD si è deciso di ricorrere ad uno zoccolo che permette un più semplice montaggio del componente e soprattutto permette di rimuovere facilmente il componente senza bisogno di dissaldare i relativi piedini.

4.1.2.3 Organizzazione dei componenti nella scheda

Una volta completata l'importazione della netlist e l'assegnazione ad ogni componente del corretto footprint, si è passati a piazzare ogni componente in una posizione che permettesse un semplice montaggio tenendo conto del package del componente. Ad esempio, i due integrati MCP1825, dovendo dissipare una certa potenza, sono stati messi vicino al bordo della scheda, per poter eventualmente permettere il montaggio di un dissipatore qualora ve ne fosse il bisogno (i.e. in caso di utilizzo della scheda in ambienti caldi). I connettori verso la MCU e verso il display sono stati piazzati ai lati opposti della scheda, per permettere il collegamento con il display e con il flat proveniente dalla scheda del microcontrollore che andrà a pilotare il driver LCD. Un'altra importante regola da seguire per individuare la posizione di un componente riguarda il fatto di mantenere vicini due componenti che abbiano molte piste in comune, questo per permettere al programma un tracciamento delle piste più semplice.

Viene di seguito riportato come appare il master con evidenziati solo i contorni dei componenti:



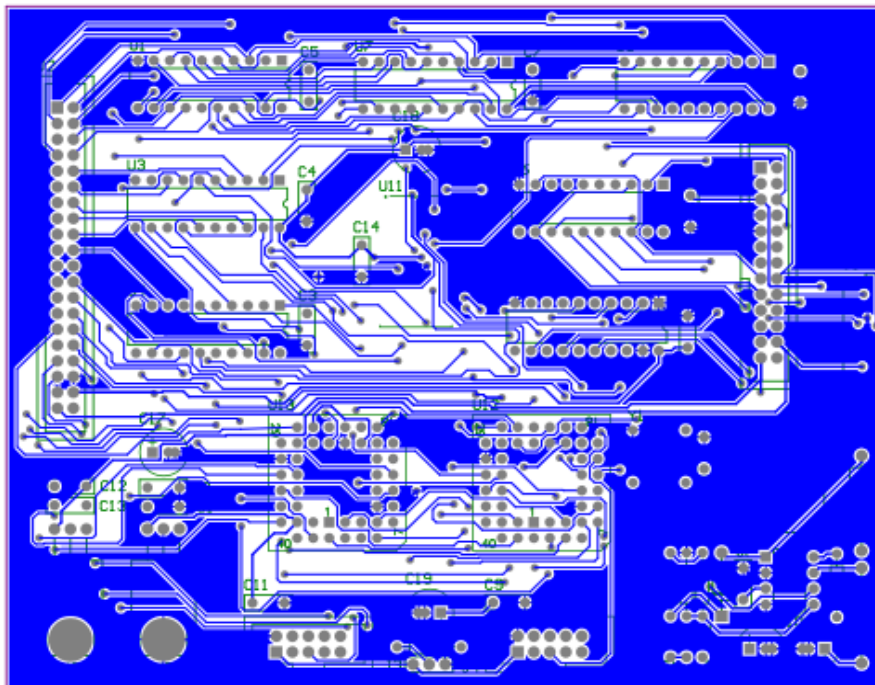
Posizione dei componenti nella scheda

Nell'immagine riportata nella pagina precedente i componenti sono indicati con il nome assunto nello schema elettrico di pagina 50.

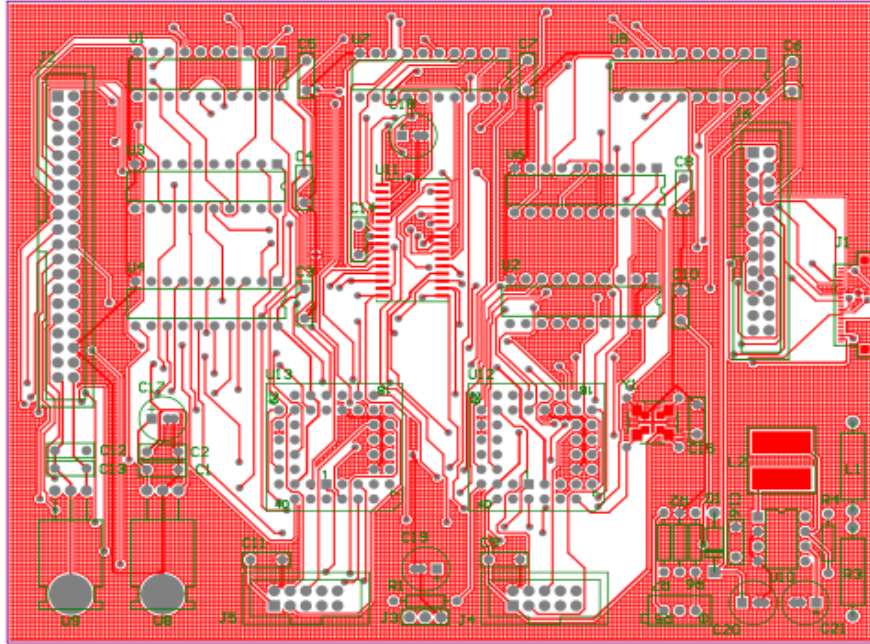
4.1.2.4 Stesura master

Una volta individuata la corretta posizione di ogni componente, ci si è affidati al programma *Altium* per quanto riguarda il tracciamento delle piste. Il programma assolve il compito seguendo delle linee guida impostate dall'utilizzatore, come la minima distanza tra due piste diverse o il tipo di collegamento tra piste (i.e. ad angolo retto o meno). Quando sono state specificate le regole per tracciare le varie net, il programma va a disegnare queste piste in maniera tale da minimizzare i ponti tra i 2 layer (top e bottom layer, cioè la superficie superiore o inferiore della scheda) e cercando di minimizzare la lunghezza delle varie interconnessioni per diminuire i disturbi che una connessione lunga può “assorbire” dall'ambiente esterno.

Avendo a disposizione due layer (*Top Layer* il superiore e *Bottom Layer* l'inferiore) in cui tracciare le interconnessioni, tramite il programma *Altium* si sono individuate i seguenti due layer:



Bottom layer



Top Layer

Come si può notare, nei due *Layer* la parte di scheda non attraversata da piste è non è stata lasciata vuota, ma è stata ricoperta da uno strato di rame collegato a massa. Quest'area di rame (*copper pour*) viene sostituita alle zone vuote e collegata a massa per permettere una migliore schermatura del circuito. Inoltre, dato che per la stampa della scheda si parte da uno strato ininterrotto di rame su cui si vanno a togliere quelle parti che permettono di formare il circuito voluto, la presenza del *copper pour* permette di diminuire la percentuale di rame da rimuovere, e quindi diminuire il tempo di lavorazione.

4.1.2.5 Stampa del prototipo

Una volta che è stato definito il master, esso viene inviato ad una ditta specializzata in produzione di schede elettriche che provvederà alla stampa del PCB secondo le specifiche fornite. In queste specifiche sono presenti tutte quelle direttive dimensionali utilizzate anche nella stesura del master, come distanza minima tra le piste o angoli di raccordo, necessarie al corretto funzionamento del circuito.

Nelle ore stabilite di tirocinio non è stato possibile completare questa fase di realizzazione, che verrà completata nella futura collaborazione del tirocinante con l'azienda.

Bibliografia

- [1] E.Sharp s.a.s. Di Borella Ing. Massimo & C. e.sharp@katamail.com
- [2] Datasheet, <http://www.displaytech.com.hk/>
- [3] Datasheet, <http://www.displaytech.com.hk/upload/product/attachment/5318-240320DP.pdf>
- [4] Datasheet, <http://ww1.microchip.com/downloads/en/DeviceDoc/22056b.pdf>
- [5] Datasheet, <http://www.onsemi.com/pub/Collateral/MC34063A-D.PDF>
- [6] Datasheet, <http://www.cypress.com/?docID=31943>
- [7] Datasheet, http://www.xilinx.com/support/documentation/data_sheets/ds057.pdf
- [8] Vogrig D., *Dispense di Elettronica dei Sistemi Digitali*. Padova, a.s. 2010/2011
- [9] Clive Max Maxfield, *The Design Warrior's Guide to FPGAs: Devices, Tools, and Flows*.
- [10] ISE Design Suite 13 www.xilinx.com
- [11] JTAG (Joint Test Action Group Standard IEEE 1149)
- [12] Applicativo, <http://www.nomad.ee/micros/mc34063a/>
- [13] www.altium.com
- [14] Forum online <http://it.narkive.com>
- [15] Display mod. 24D00 www.edt.com