



Università degli Studi di Lecce
Facoltà di Ingegneria
Corso di Laurea in Ingegneria dell'Informazione

Tesi di Laurea

SVILUPPO DI UN DISPOSITIVO PER LA SINCRONIZZAZIONE DI SISTEMI DI MISURA REMOTI

Relatore: Chiar.mo Prof. Marco Panareo

Laureando: Giuseppe Cocciolo

Anno Accademico 2004 / 05

*Alla mia famiglia,
che nei momenti di difficoltà
mi ha sostenuto, consentendomi
di proseguire con costanza e
impegno la strada intrapresa.*

Ringrazio il Prof. Marco Panareo, per avermi permesso di svolgere la tesi presso l'Istituto Nazionale di Fisica Nucleare e per il compito di relatore svolto con grande impegno.

Un altro ringraziamento è rivolto, anche, al gruppo di lavoro del Laboratorio di Elettronica dell' INFN, in particolare al Dott. Alessandro Corvaglia, che, con il suo contributo costante e prezioso, mi ha aiutato a superare le difficoltà che, volta per volta si sono presentate.

INDICE

Introduzione	1
CAPITOLO 1	
1.1 L'esperimento EEE	3
1.2 Il rivelatore	3
1.3 Il telescopio a MRPC	4
CAPITOLO 2	
Il sistema GPS	
2.1 Cenni storici	6
2.2 Parti di cui è costituito il sistema	7
2.3 Orario del sistema GPS	9
2.4 Frequenze utilizzate e codici trasmessi	10
2.5 Ricezione ed utilizzo dei segnali	11
2.5.1 Ricevitori GPS	11
2.5.2 Misura dei ritardi e calcolo delle distanze	12
2.6 Errori del sistema	12
2.7 Il segnale PPS	13
2.8 Interfaccia dati: NMEA – 0183	14
2.9 Struttura del protocollo NMEA	14
CAPITOLO 3	
3.1 Teoria dell'operazione	17
3.2 Programmazione della PIC	19
3.3 test del programma	24
3.4 Sviluppo successivo del programma	26
3.5 Sviluppo del programma per le XC9572-PC84	27
3.6 Simulazione al PC	32
3.7 Test del programma	34
3.8 Programma generale da realizzare per la ricezione delle informazioni	36
3.9 Simulazione al PC	38
3.10 Sviluppo del programma per le Spartan XCS10PC84	42
3.11 Simulazione al PC e test del programma	45
3.129 Sviluppo del programma per la gestione del tempo fino ai ns	47
3.13 Fonti di errore	50
3.14 Simulazione al PC e test del programma	52

Conclusioni	53
Appendice A Misure effettuate	54
Appendice B Programma della PIC	57
Appendice C Programma in LabView	62
Bibliografia	65

INTRODUZIONE

L'obiettivo della tesi è la costruzione di un dispositivo, che consente di estrarre le informazioni (con particolare riguardo ai dati temporali) fornite da un ricevitore GPS, parallelizzarle e renderle disponibili su richiesta verso dispositivi esterni.

Tale dispositivo è utilizzato nell'ambito dell'esperimento EEE (Extreme Energy Events), che ha come oggetto la rivelazione dei raggi cosmici di alta energia. Una particella di origine cosmica, interagendo con l'atmosfera terrestre, produce altre particelle, dando luogo ad uno sciame (**figura 1**).

L'estensione spaziale dello sciame dipende dall'energia della particella primaria, così per poter osservare particelle di elevata energia occorre disporre di rivelatori distribuiti in una vasta area geografica. L'esperimento EEE prevede l'installazione di circa 100 rivelatori di particelle presso altrettanti istituti scolastici italiani.

L'angolo di incidenza della particella primaria nell'atmosfera può essere dedotto attraverso una misura delle differenze tra i tempi di arrivo delle particelle su ogni rivelatore. Ciò richiede che ogni stazione di misura sia dotata di precisi orologi, sincronizzati tra loro entro pochi nanosecondi.

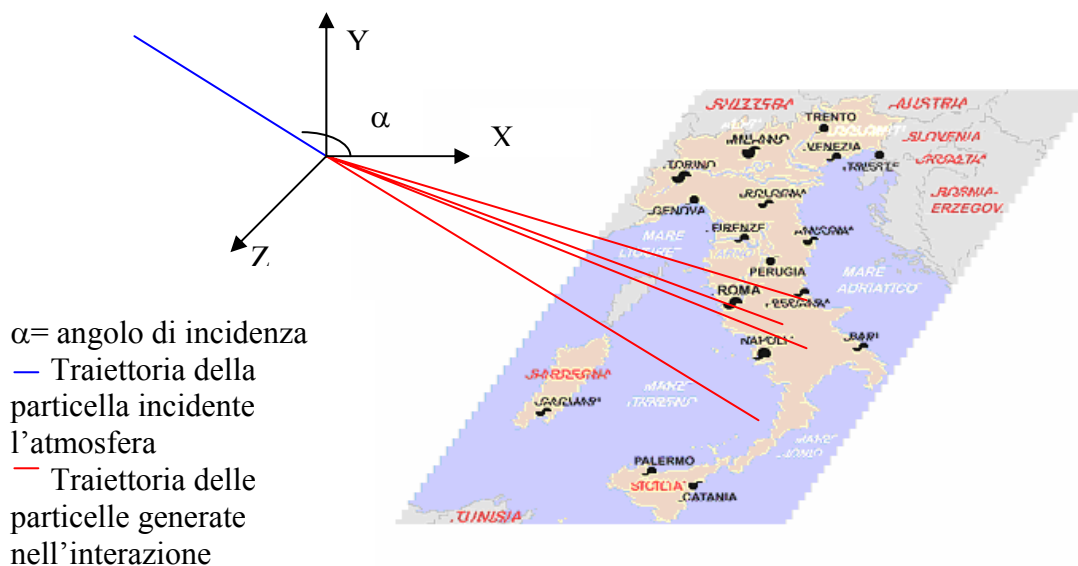


Figura 1

Per questo è venuta l'idea di utilizzare, come sistema di sincronizzazione, il sistema GPS (Global Position System), che, grazie agli orologi atomici di elevata precisione di cui è dotato, fornisce indicazioni temporali estremamente accurate.

I dispositivi che costituiscono il modulo costruito sono: un microcontrollore, nel nostro caso è stata utilizzata una PIC (Peripheral Interface Controller) che ha il compito di estrarre i dati inviati serialmente da un modulo commerciale

GPS di alta precisione e da un PLD (Programmable Logic Device), utilizzato per parallelizzare le informazioni fornite dalla PIC, al fine di poterle interfacciare con un bus standard.

Queste operazioni vengono eseguite quando una particella attraversa un rivelatore, il quale genera un segnale che ha il compito di abilitarne l'esecuzione.

La parte del progetto sviluppata nella tesi è quella di:

- 1) Costruire un algoritmo per la programmazione della PIC e il relativo programma che lo implementa.
- 2) Costruire un prototipo per il test, effettuato con un analizzatore di stati logici.
- 3) Sviluppare gli algoritmi che saranno implementati nelle PLD. Il test preliminare di questi dispositivi, avviene mediante simulazione al PC.
- 4) Costruire un prototipo per il loro test effettuato tramite l'analizzatore di stati logici.
- 5) Sviluppare un algoritmo per una PLD, in grado di fornire l'informazione temporale dell'ordine dei ns , con relativa simulazione al calcolatore.

Questa tesi è stata svolta col supporto del Centro Studi e Ricerche *Enrico Fermi* di Roma e della Sezione di Lecce dell'Istituto Nazionale di Fisica Nucleare.

CAPITOLO 1

1.1 L'esperimento EEE

L'esperimento EEE ha come obiettivo lo studio della radiazione di origine cosmica di elevata energia. Quando un protone cosmico incontra l'atmosfera terrestre, interagisce con i nuclei di questa, producendo delle particelle secondarie. Il tempo di vita di queste è molto breve, dopo di che decadono in altre particelle, tra cui i muoni, che costituiscono la maggior parte della componente elettricamente carica dei raggi cosmici, a livello del mare,. I raggi cosmici primari di altissima energia generano sciame estesi, composti da un elevato numero di particelle (legato all'energia del protone primario). L'area, coperta da uno sciame prodotto da una particella ad altissima energia, può arrivare ad alcune decine di chilometri quadrati.

1.2 Il rivelatore

I rivelatori usati in questo esperimento prendono il nome di MRPC (Multigap Resistive Plate Chamber). Ogni rivelatore consiste in una pila ($1,6 \times 0,81 \text{ m}^2$) di lastre di vetro (di spessore 2mm), separate $350 \text{ }\mu\text{m}$ l'una dall'altra, comprese tra due lastre di vetro rese parzialmente conduttrici attraverso una opportuna verniciatura. Il sistema è racchiuso in un contenitore stagno in alluminio ed è riempito di Freon. Tra le due lastre conduttrici è applicata una differenza di potenziale di circa 8 KV (**figura 2**).

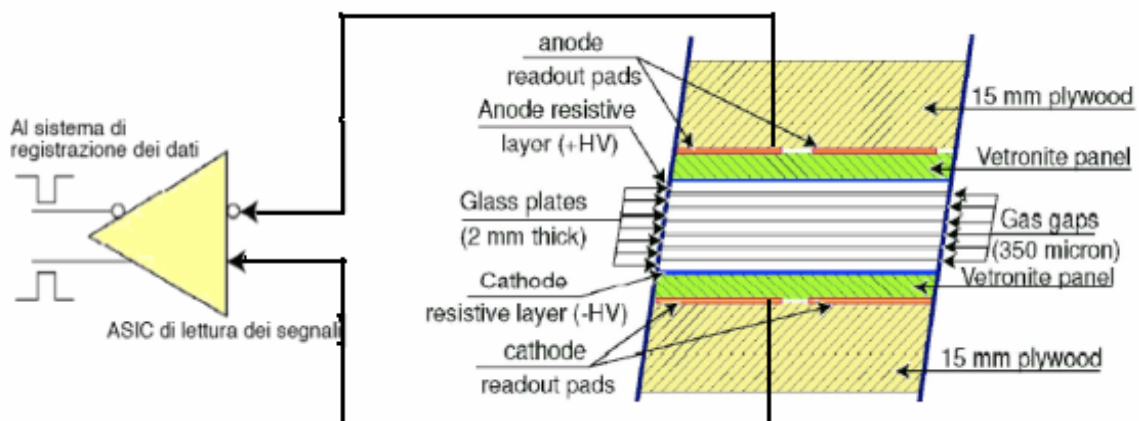


Figura 2 Principio di funzionamento di un Rilevatore MRPC

Una particella carica, che attraversa il gas, se ha l'energia sufficiente ionizza le molecole presenti lungo la sua traiettoria. Le cariche elettriche generate nel gas all'interno del rivelatore si muovono verso le lastre conduttrici: gli elettroni verso quella carica positivamente, gli ioni positivi verso quella carica negativamente. Il movimento di queste cariche verso le armature dà origine a un

segnale elettrico che viene indotto su un sistema di elettro metallici costituiti da strisce longitudinali, ciascuna lunga $1,6m$ e larga $34mm$. Ogni striscia è connessa, a ciascuna delle sue estremità, con un sistema elettronico di lettura e di acquisizione del segnale. La differenza in tempo tra i segnali di due estremi di ogni striscia fornisce la coordinata longitudinale del punto d'impatto con la precisione di circa $1cm$. Gli elettrodi a striscia sono posizionati all'interno del contenitore metallico, all'esterno delle lastre conduttrici di parallelizzazione. Questo segnale elettrico viene usato per rivelare il passaggio della particella.

1.3 Il telescopio con MRPC

Il sistema di rivelazione del progetto è un telescopio costituito da tre piani di rivelatori MRPC (**figura 3**).

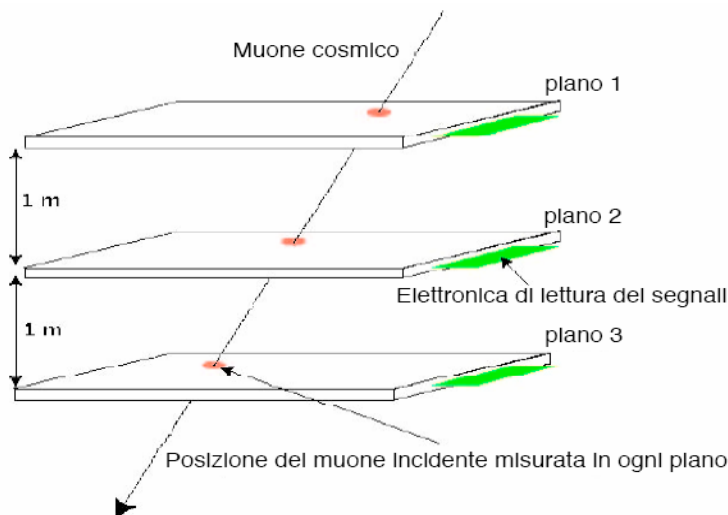


Figura 3 Telescopio con MRPC

Ogni piano è capace di misurare con grande precisione il punto d'impatto della particella cosmica incidente e il suo tempo di attraversamento.

Tramite la misura dei tre punti d'impatto (uno per piano), è possibile ricostruire la traiettoria della particella che ha attraversato il telescopio. È inoltre possibile conoscere il verso di attraversamento della particella, grazie alla misura del suo tempo di volo tra un piano e l'altro.

Per la lettura e l'acquisizione dei dati a ogni telescopio sarà associata una catena elettronica costituita da: un sistema per l'amplificazione e la discriminazione dei segnali forniti dagli elettrodi di lettura dei rivelatori MRPC; un sistema di conversione, per la digitalizzazione delle informazioni acquisite e da un sistema detto di trigger per la selezione delle particelle.

In particolare, quest'ultimo sistema è fatto in modo da generare un segnale (trigger), quando almeno una striscia di ogni singolo piano viene attraversata da una particella. Il segnale di trigger si ottiene effettuando un OR logico fra tutti i segnali provenienti dalle strisce di ogni singolo piano e, successivamente, ponendo i tre OR in AND logico. Queste operazioni, avvengono in una definita finestra temporale, cioè il livello logico delle OR viene mantenuto per il periodo di tempo necessario a far sì che la particella attraversi tutti e tre i piani, in modo tale che l'AND logico possa avere, per un istante di tempo sufficiente, tutte le linee d'ingresso a livello alto, e quindi generare il segnale.

Tramite un'opportuna interfaccia la catena elettronica è connessa con un calcolatore; ogni telescopio è, quindi, in grado di acquisire dati e di trasmetterli via rete ad un opportuno centro di raccolta. Tutti i telescopi, dei quali si conosce l'esatta posizione geografica, essendo temporalmente sincronizzati via satellite tramite un sistema GPS, sono messi in coincidenza durante la fase di analisi dei dati. Questo consente di rivelare eventi cosmici di energie elevate che si manifestano attraverso sciame cosmici di grande apertura angolare, ognuno dei quali costituito da un notevole numero di muoni, provenienti da un punto comune nell'alta atmosfera terrestre.

Il rivelatore MRPC deve garantire un'ottima precisione per misurare i tempi di volo delle particelle cosmiche e le loro traiettorie, necessarie per stabilire il punto, in cui si è verificata nell'atmosfera l'interazione che ha prodotto la serie di eventi registrati dai rivelatori.

CAPITOLO 2

IL SISTEMA GPS

2.1 Cenni storici

Il sistema GPS è un sistema satellitare basato su una costellazione di 24 satelliti orbitanti ad una quota di circa 20200 Km e quindi, con un periodo orbitale di 12 ore sideree ($11^h 58^m 02^s$).

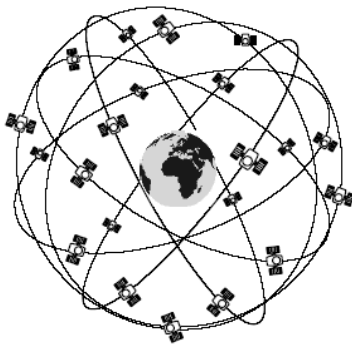


Figura 4 Costellazione dei satelliti

Esso rappresenta un sistema di navigazione globale, continuo e tridimensionale, in grado di fornire, con estrema precisione, le coordinate geografiche, la quota e la velocità di qualsiasi mezzo in ogni punto della Terra e per l'intero arco delle ventiquattro ore.

Il programma per la realizzazione del sistema è stato creato dal dipartimento della difesa degli

USA nel 1973 per consentire a navi, aeroplani e veicoli terrestri delle forze armate degli USA di determinare, in qualsiasi istante, la propria

posizione in qualunque parte del globo si trovino. Pur essendo stato concepito per scopi militari, i progettisti del sistema ne hanno consentito l'uso anche ai civili. I primi 11 satelliti di tipo sperimentale sono stati lanciati dal 1978 al 1985 e sostituiti, successivamente, da quelli operativi, a partire dal 1989.

L'interesse da parte della comunità scientifica per il sistema e gli studi condotti hanno consentito agli apparati riceventi di utilizzarlo con una precisione notevolmente superiore a quella ipotizzata dai progettisti. Per tale motivo le applicazioni pratiche nel settore civile sono diventate numerosissime anche al di fuori dei campi tradizionali della navigazione marittima ed aerea, ad esempio:

- L'estrema precisione del sistema consente il controllo della posizione di automezzi terrestri e, precisamente, un ricevitore GPS determina con continuità la posizione che viene comunicata via radio ad una centrale di controllo.
- L'introduzione del DGPS o GPS differenziale ha consentito di raggiungere precisioni adeguate per poterlo utilizzare come strumento di posizione, sia statico che dinamico, nelle misure geodetiche, geofisiche, idrografiche e cartografiche.

Le applicazioni elencate sono soltanto alcune di quelle in uso, molte altre sono in fase studio, per cui il sistema avrà un'importanza sempre maggiore in tutti i campi della vita civile.

2.2 Parti di cui è costituito il sistema

Il sistema è diviso in tre segmenti:

- 1) **Segmento spaziale:** è formato da una costellazione nominale di 24 satelliti di cui 21 sono attivi e 3 sono di riserva, pronti per essere utilizzati quando vi è la necessità, come nel caso in cui qualche satellite principale entri in avaria oppure per garantire una migliore copertura in alcune aree terrestri in particolari occasioni (prevalentemente per scopi militari).

Ciascun satellite trasmette con continuità degli opportuni segnali codificati che contengono diverse informazioni fra cui i dati orbitali per il calcolo della posizione del satellite ed un segnale di tempo per la determinazione degli istanti di partenza dei suddetti segnali.

I satelliti (**figura 5**) sono distribuiti su 6 diversi piani orbitali inclinati di 55° rispetto all'equatore e con una differenza di 60° in longitudine tra i nodi ascendenti. I quattro satelliti di ciascun gruppo sono distribuiti uniformemente sul piano orbitale d'appartenenza, tuttavia tale collocazione può essere variata dal centro di controllo terrestre quando sussistono particolari esigenze.

La quota è stata scelta in modo che il periodo orbitale sia nominalmente di 12 ore sideree e, quindi, con due orbite complete descritte in un giorno sidereo ($23^h 56^m 04^s$). La quota prevista, di circa 20200 Km, consente un'adeguata visibilità dei satelliti da parte degli utilizzatori terrestri.

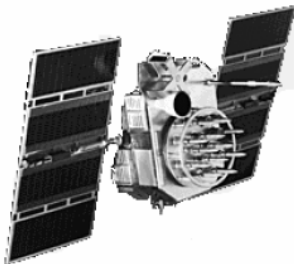


Figura 5

La costellazione dei 24 satelliti è stata disegnata, in modo tale da consentire la presenza di almeno 4 di essi sopra l'orizzonte di una qualsiasi località

terrestre e con un'elevazione superiore a 5° . Tale copertura garantisce inoltre un'adeguata distribuzione spaziale dei satelliti anche nel caso in cui qualcuno di essi sia momentaneamente fuori servizio o disattivato; in tal modo è sempre possibile effettuare la navigazione tridimensionale (latitudine, longitudine e quota) e la sincronizzazione degli orologi di bordo con un sufficiente grado di precisione.

- 2) **Segmento di controllo:** ha la funzione di seguire il moto orbitale dei satelliti ed il funzionamento dei loro orologi e, quindi, di aggiornare continuamente i dati conservati nelle memorie di bordo. Il sistema è



Figura 6 Stazioni di controllo terrestre

composto da cinque stazioni situate intorno al globo in maniera pressoché uniforme; il centro operativo del sistema MSC (Master Station Control) è situato a Colorado Spring negli Stati Uniti. Tutte e cinque le stazioni svolgono la funzione di stazioni monitor MS (Monitor Stations) ossia effettuano con continuità ed in modo automatico la misura della loro distanza dai satelliti e di altri dati utili per valutare il corretto funzionamento degli apparati di bordo. I dati rilevati sono immagazzinati ed inviati alla MSC su sua specifica richiesta; quest'ultima provvede a calcolare i dati orbitali di tutti i satelliti, le correzioni degli orologi collocati a bordo ed altri dati destinati a far parte del messaggio che verrà inoltrato ai singoli satelliti dalle stazioni di aggiornamento ULS (Upload Stations). Il caricamento dei dati avviene, in condizioni normali, ogni otto ore su una frequenza di servizio.

Il corretto utilizzo del sistema richiede una grande precisione nella determinazione del tempo, pertanto la MSC è dotata di una serie di orologi atomici estremamente precisi ai quali vengono riferiti tutti gli altri orologi, sia a terra che a bordo dei satelliti.

Tempo e frequenza a bordo dei satelliti sono forniti da tre o quattro orologi atomici (sia al Cesio che al Rubidio) sincronizzati giornalmente dal centro di controllo terrestre per compensare le derive e gli errori, che si manifestano col trascorrere del tempo.

- 3) **Segmento dell'utilizzatore:** consiste in una varietà di ricevitori, di decodificatori e di elaboratori dei segnali GPS.

I ricevitori terrestri, mediante un proprio orologio interno, misurano la differenza fra l'istante di ricezione e quello di trasmissione del segnale; moltiplicando tale differenza per la velocità di propagazione delle onde elettromagnetiche si ottiene la distanza fra il satellite ed il ricevitore.

Il luogo di posizione così definito è una sfera con il centro nella posizione occupata dal Satellite nell'istante di emissione del segnale ed il raggio pari alla distanza calcolata; tale luogo interseca la superficie terrestre, formando una circonferenza che è il luogo dei punti in cui si trova l'utilizzatore terrestre (**figura 7**). Con due misure di distanza e quindi utilizzando due satelliti, si individuano due circonferenze che si intersecano in due punti di cui uno è la posizione dell'osservatore; l'ambiguità fra i due punti può essere eliminata con la posizione stimata del ricevitore.

Considerando come incognita anche la quota diventano necessarie tre osservazioni che individuano tre sfere la cui intersezione individua un volume entro il quale si trova il ricevitore. Nel caso bidimensionale, ossia nel caso in cui la quota è nota, le tre misure individuano un triangolo praticamente piano, la cui area è funzione degli errori di misura e della distribuzione dei satelliti (**figura 7**). È fondamentale disporre di orologi di grande precisione nell'ambito di tutto il sistema, per cui i satelliti sono dotati di orologi atomici, che sono costantemente controllati dal MSC. Gli orologi installati negli apparati ricevitori terrestri non devono, necessariamente, avere una grande precisione, essendo possibile determinare, costantemente, il loro errore rispetto al tempo simultaneo, fornito dai satelliti. L'errore dell'orologio rappresenta, pertanto, un'ulteriore incognita, che può essere determinata con un'ulteriore osservazione; si risolve, in pratica, un sistema di quattro equazioni in quattro incognite: latitudine, longitudine, quota ed errore dell'orologio.

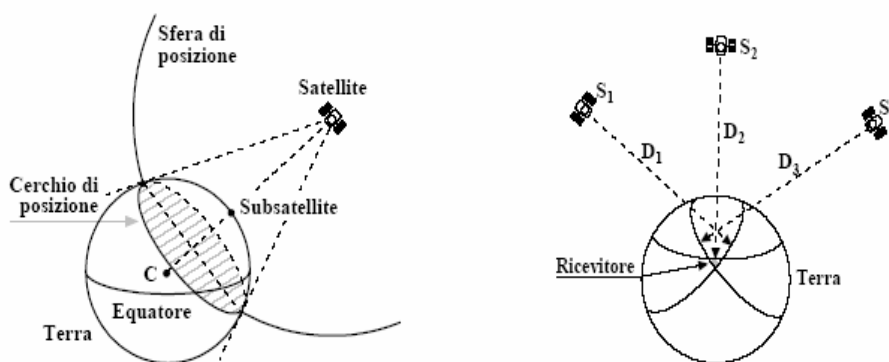


Figura 7

2.3 Orario del sistema GPS

Inizialmente il BIH (Bureau International de l'Heme), operante presso l'Osservatorio di Parigi, calcolava, in base alle osservazioni effettuate dai laboratori astronomici sparsi in tutto il mondo, l'UT1, la scala di riferimento, basata sulla rotazione terrestre.

Successivamente, con la disponibilità di orologi più accurati è stata realizzata la scala UTC, che si basa sulla corrente definizione del secondo fornita dal Sistema Internazionale (Il secondo è la durata di 9.192.631.770 periodi della radiazione corrispondente alla transizione tra due livelli iperfini dello stato fondamentale dell'atomo di Cesio-133). Il primo gennaio 1968 è stato sincronizzato l'UTC con l'UT1. Poichè le differenze tra l'UTC e l'UT1, con il passare del tempo, aumentavano (questo a causa della irregolare rotazione della terra), nel gennaio 1972 è stato deciso di introdurre nella scala UTC il secondo intercalare (periodico aggiustamento della scala UTC, che consiste nell'aggiunta di un secondo per mantenere essa in stretto contatto con la scala UT1), che viene inserito alla fine di giugno o di dicembre.

È stata, inoltre, creata una scala, la TAI (International Atomic Time), utilizzando i dati riguardanti l'UTC raccolti dal BIPM (Bureau International des Poids et Mesures) e provenienti da 230 orologi atomici, situati in 60 laboratori, sparsi in tutto il mondo. Sottraendo a tali dati il valore dell'UTC proveniente dal BIPM, si ottiene, mediante media ponderata del risultato, il TAI.

Mensilmente il BIPM produce un bollettino ufficiale, dove vengono evidenziati gli scarti tra ogni laboratorio e l'UTC.

Il riferimento temporale utilizzato nel sistema GPS, viene dato dal GPS Time, non potendo essere utilizzato l'UTC, a causa dell'aggiunta del secondo intercalare.

2.4 Frequenze utilizzate e codici trasmessi

Tutti i satelliti trasmettono le informazioni, utilizzando due diverse portanti nella banda L, che sono entrambe multiple di una frequenza fondamentale $f_0 = 10,23$ MHz, generata da una serie di oscillatori montati a bordo. Indicando con L_1 e L_2 le portanti, si ha:

$$L_1 = 154 \times f_0 = 1575,42 \text{ MHz}$$

$$L_2 = 120 \times f_0 = 1227,60 \text{ MHz}$$

Le due portanti sono modulate in fase con appositi segnali denominati:

- Codice P (Precision).
- Codice C/A (Coarse Acquisition).
- Codice D (Navigation Data).

Tali codici consistono in sequenze di cifre binarie, perfettamente note e ripetute ad intervalli di $1ms$ per il codice C/A e di 7 giorni per il codice P.



Figura 8 Esempio di una sequenza di bit; nel caso del codice C/A essa è formata da 1023 bit, per una durata complessiva di $1ms$. Nel caso del codice P la sequenza è di ~ 15 milioni di BIT per una durata di una settimana.

Ogni satellite utilizza una propria sequenza, differenziata da quelle degli altri, grazie ad un numero variabile da 0 a 37, che viene indicato con la sigla PRN (Pseudo Random Noise) (**figura 9**).



Figura 9 Codice P che si ripete periodicamente ogni 38 settimane, esso è stato diviso in segmenti settimanali, ed ognuno di essi è stato assegnato ad un diverso satellite.

Fin dall'inizio della fase sperimentale, è stato accertato, che la precisione del GPS, utilizzando anche soltanto il codice C/A, può essere nettamente superiore a quella prevista dai progettisti del sistema. In base alle specifiche del Federal Radionavigation Plan del Governo USA, sono previsti due livelli di utilizzo:

- **Standard Positioning Service (SPS)** che utilizza il codice C/A sulla portante L_1 , con accuratezza di 100 m nel piano orizzontale.
- **Precise Positioning Service (PPS)** che utilizza il codice P su entrambe le portanti L_1 e L_2 , con un'accuratezza di 18 m nel piano orizzontale.

2.4 Ricezione ed utilizzo dei segnali

2.4.1 Ricevitori GPS

Un ricevitore GPS comprende essenzialmente quattro parti: l'antenna, il ricevitore, un microprocessore ed i dispositivi per l'introduzione e la visualizzazione dei dati. Le funzioni fondamentali di un ricevitore possono essere sintetizzate come segue:

- Selezionare quattro satelliti in base ai dati contenuti nel messaggio di navigazione, cioè alle effemeridi approssimate di essi, che consentono al ricevitore di determinare velocemente la loro posizione, in modo da stabilire quali siano visibili ed utilizzabili, al fine di consentire la migliore determinazione della posizione.
- Individuare i segnali dei satelliti selezionati, decodificandoli attraverso il proprio PRN.

- Decodificare i dati di navigazione contenuti nel codice D e memorizzarli nella propria memoria.
- Misurare i ritardi nei tempi d'arrivo dei segnali dai satelliti e, calcolare le relative distanze.
- Calcolare la posizione del ricevitore e l'ora del sistema (GPS Time).

2.4.2 Misura dei ritardi e calcolo delle distanze

Il codice C/A arriva all'utilizzatore con un ritardo $\Delta t = R/c$, dove R è la distanza fra il satellite ed il ricevitore e $c=300000$ km/sec è la velocità di propagazione delle onde elettromagnetiche. Il ricevitore è dotato di un generatore di codice C/A, analogo a quello di cui sono dotati i satelliti e, come questi ultimi, dovrebbe essere sincronizzato con il GPS Time. Tuttavia, tale sincronismo non è perfetto, essendo utilizzati nei ricevitori degli orologi, normali oscillatori al quarzo, che hanno una sensibile deriva di cui occorre tenere conto nella misura dei tempi.

I due segnali, quello proveniente dal satellite e quello generato internamente al ricevitore, vengono confrontati con una tecnica detta di autocorrelazione dei segnali; il dispositivo elettronico che attua tale confronto è chiamato correlatore e da in uscita un segnale $V(\tau)$ che è funzione dello sfasamento fra i due segnali in ingresso. Più precisamente, $V(\tau)$ assume il valore massimo, quando i due segnali sono perfettamente allineati ed un valore nullo o molto piccolo nel caso di segnali sfasati fra di loro.

2.5 Errori del sistema

Il sistema GPS è soggetto a differenti tipi di errori, che influiscono sulla distanza misurata; molti di essi possono essere ridotti di ampiezza, utilizzando delle formule, ottenute dopo un'attenta valutazione delle cause e dopo una serie di misure effettuate durante la fase sperimentale del sistema. Fondamentalmente essi sono:

- **Errori dipendenti dai satelliti**, fra cui anche la **Selective Availability** e l'**AntiSpoofing**. Questi sono dispositivi introdotti, inizialmente, dal dipartimento della difesa degli USA, a bordo dei satelliti per creare artificialmente una degradazione della precisione (la massima precisione era garantita, soltanto, agli utenti autorizzati) e per proteggere gli utilizzatori militari dal pericolo di essere ingannati da false trasmissioni con segnali simili a quelli del GPS. Recentemente tali dispositivi sono stati disattivati, consentendo ai ricevitori civili un incremento della precisione.
- **Errori dipendenti dal ricevitore**. Ogni ricevitore genera degli errori legati al rumore interno, alla precisione con cui lavora il correlatore, ai ritardi prodotti sia dai vari dispositivi elettronici che dal software che

effettua l'elaborazione dei dati. Gli effetti di tali errori possono essere consistenti nel caso di ricevitori utilizzati su veicoli molto veloci; una progettazione accurata del ricevitore in base all'uso cui è destinato può pertanto renderli molto piccoli.

- **Errori prodotti dalle riflessioni multiple.** Essi derivano, principalmente, dalla combinazione dei segnali diretti con quelli riflessi dalle superfici circostanti, in particolare dalla superficie marina. Tali errori sono, quindi, dipendenti dalla natura e dalla localizzazione delle superfici riflettenti, per cui è possibile ridurli o eliminarli con un'opportuna collocazione e progettazione dell'antenna ed adottando opportune tecniche nei ricevitori.
- **Errori prodotti dalla propagazione dei segnali nella ionosfera e nella troposfera.** Nell'attraversamento di tali strati, le onde elettromagnetiche subiscono sia variazioni nella velocità sia delle rifrazioni che producono un allungamento dei percorsi rispetto a quelli rettilinei fra i satelliti e il ricevitore.
- **Errori introdotti dal Sistema di Controllo terrestre** nella determinazione delle orbite, nelle correzioni degli orologi e così via.

Tutti questi errori rendono incerti i luoghi di posizione individuati e quindi la posizione finale del ricevitore. L'errore sulla posizione dipende, oltre che dagli errori nelle misure, da un fattore legato alla geometria del sistema, ossia alla distribuzione nello spazio dei satelliti utilizzati nelle misure; tale fattore è denominato fattore d'espansione dell'errore o PDOP (Position Dilution Of Precision).

Nel caso della navigazione marittima o terrestre interessa soltanto la posizione e quindi la precisione nel piano orizzontale, per cui il PDOP viene sostituito dall'HDOP (Horizontal DOP). I ricevitori, utilizzando i dati contenuti nel messaggio di navigazione sono tuttavia in grado di calcolare, preventivamente, il PDOP o l'HDOP per le varie combinazioni di satelliti visibili e di scegliere, quindi, la combinazione migliore con il fattore più basso.

2.6 Il segnale PPS

L'utilizzo del GPS permette di accedere, a basso costo dal proprio laboratorio, alla stabilità di un insieme di orologi atomici. In effetti, attraverso un'opportuna scelta del ricevitore GPS, chiunque, in ogni luogo, può non solo ottenere in tempo reale la conoscenza delle proprie coordinate geografiche, come la latitudine e la longitudine, ma anche ricevere un particolare segnale ogni secondo, il cosiddetto one pulse per second (1PPS), sincronizzato con il tempo UTC.

Se l'antenna ricevente è mantenuta in una posizione fissa, come in un laboratorio, il segnale 1PPS può essere utilizzato come sistema di riferimento per il tempo. I moderni ricevitori destinati ad applicazioni per la misura del tempo possono prendere in considerazione quasi tutte le cause di ritardo del sistema, in maniera tale che il segnale 1PPS sia sincronizzato entro pochi nanosecondi con il tempo UTC e reso disponibile localmente.

Il segnale 1PPS, fornito dai ricevitori GPS, è in assoluto il migliore standard disponibile per lunghi tempi d'integrazione.

2.7 Interfaccia dati: NMEA – 0183

Il GPS comunica i dati tramite uno speciale formato, standardizzato dal National Marine Electronics Association (NMEA). I seguenti insiemi di dati sono utilizzati dal GPS per trasmettere le informazioni in formato NMEA:

1. GGA (dati fissi per il Global Position System)
2. GLL (posizione geografica – Latitudine/Longitudine)
3. GSA (degradazione di accuratezza e il numero di satelliti attivi nel sistema di navigazione satellitare globale (GNSS))
4. GSV (satelliti in vista nel GNSS)
5. RMC (minimo GNSS raccomandato)
6. VTG (direzione, velocità orizzontale)
7. ZDA (tempo e data)

2.8 Struttura del protocollo NMEA

La velocità con cui i dati vengono trasmessi nello standard NMEA è 4800 Baud, usando caratteri ASCII a 8 bit. La trasmissione inizia con un bit di start (zero logico), seguito da 8 bit di dati e un bit di stop (uno logico), aggiunto alla fine. Non è usato il bit di parità.

Ogni set di dati, provenienti dal GPS, è formato nello stesso modo e ha la seguente struttura:

\$GPDTS,inf_1,inf_2,inf_3,inf_n*CS<CR><LF>.

Le funzioni dei caratteri individuali o dei set di caratteri è spiegata in **Tabella 1**.

Tabella 1

Campo	Descrizione
\$	Inizio del set di dati
GP	Informazioni provenienti dal GPS
DTS	Identificatore del set di dati
inf_1...inf_n	Informazioni contenute nel set
,	Separatore fra le informazioni
*	Separatore per il checksum
CS	Checksum (parola di controllo)

<CR><LF>	Fine del set di dati
----------	----------------------

Il numero massimo di caratteri usati non può essere superiore a 79. Nel conteggio del numero di caratteri non vengono inclusi il carattere di inizio \$ e quelli di fine <CR><LF>.

Le seguenti stringhe sono ottenute dalla ricezione di un messaggio proveniente dal GPS in formato NMEA:

Tabella 2

\$GPRMC,130303.0,A,4717.115,N,00833.912,E,000.03,043.4,200601,01.3,W*7D<CR><LF>
\$GPZDA,130304.2,20,06,2001,,*56<CR><LF>
\$GPGGA,130304.0,4717.115,N,00833.912,E,1,08,0.94,00499,M,047,M,,*59<CR><LF>
\$GPGLL,4717.115,N,00833.912,E,130304.0,A*33<CR><LF>
\$GPVTG,205.5,T,206.8,M,000.04,N,000.08,K*4C<CR><LF>
\$GPGSA,A,3,13,20,11,29,01,25,07,04,,,,,1.63,0.94,1.33*04<CR><L>
\$GPGSV,2,1,8,13,15,208,36,20,80,358,39,11,52,139,43,29,13,044,36*42<CR><LF>

Nella tesi sono prese in considerazione solo quelle identificate da GGA, GSA, RMC, poiché esse contengono tutte le informazioni di nostro interesse. Esaminiamo una di esse come esempio:

Tabella 3

Campo	Descrizione
\$	Inizio del set di dati
GP	Informazioni originate dal GPS
GGA	Identificatore del set di dati
130304.0	UTC 13h 03min 04.0sec
4717.115	Latitudine: 47° 17.115 min
N	Latitudine da nord (N = nord, S = sud)
00833.912	Longitudine: 8° 33.912min
E	Longitudine da est (E=est, W=ovest)
1	Qualità dettagliGPS(0=no GPS, 1=GPS, 2=DGPS)
08	Numero di satelliti usati
0.94	Diluzione di precisione orizzontale (HDOP)
00499	Altezza antenna
M	Unità di misura dell'altezza (M=metri)
047	Differenza di altezza tra geoide e ellissoide
M	Unità di misura differenza di altezza (M=metri)
,,	Età dei dati DGPS (non è usato DGPS)

0000	Identificatore della stazione di riferimento del DGPS
*	Separatore per il checksum
58	Checksum per la verifica dell'intero set di dati
<CR><LF>	Fine del set di dati

CAPITOLO 3

3.1 Teoria dell'operazione

In **figura10** è riportato lo schema a blocchi delle operazioni che il modulo progettato deve effettuare.

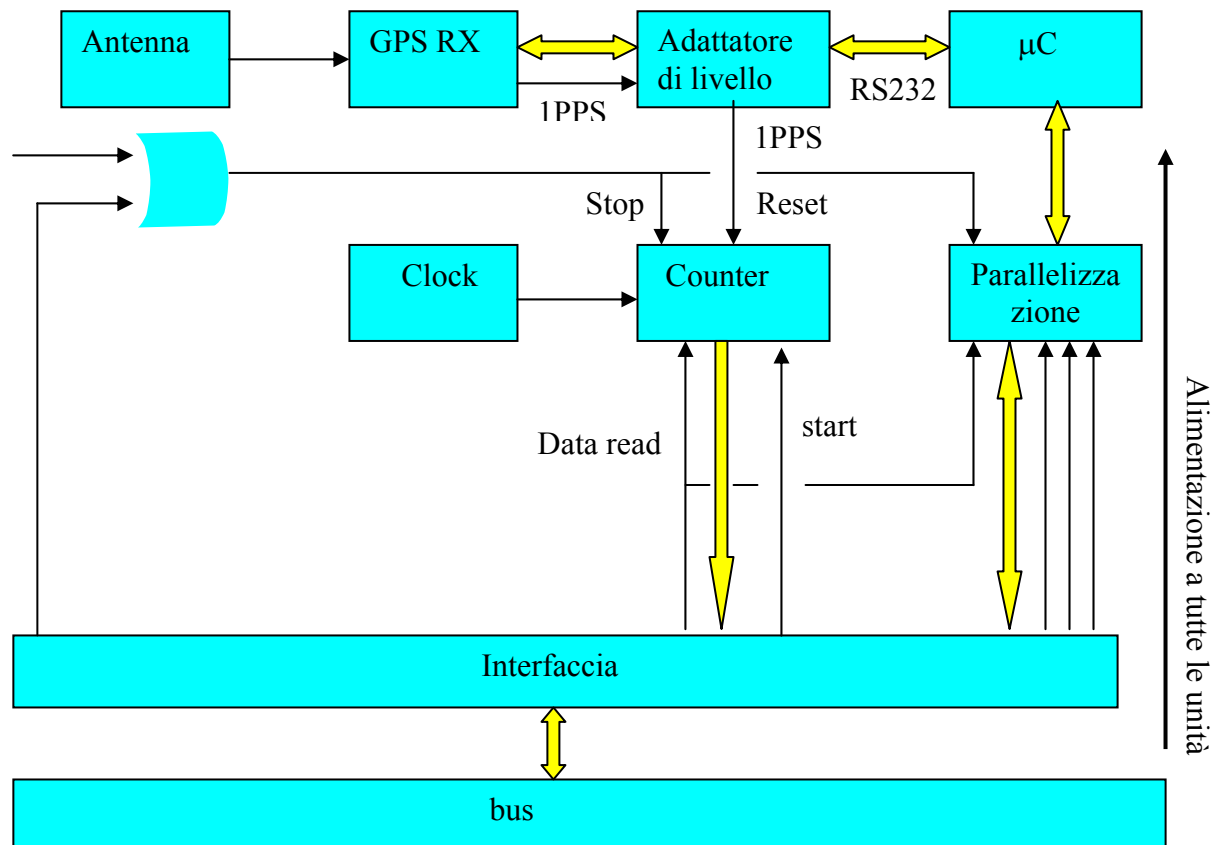


Figura 10 Schema a blocchi

I primi due, l'antenna e il ricevitore GPS sono costituiti da dispositivi esterni al modulo, mentre, i blocchi Adattatore di livello, μC , Clock, Counter, Parallelizzazione, sono realizzati con dispositivi interni.

Il GPS invia i dati in formato NMEA e il segnale 1PPS, verso il blocco adattatore di livello, costituito dal MAX RS232, il cui compito è quello di convertire le stringhe di dati dallo standard TTL a quello RS232.

Una volta convertite, le stringhe vengono inviate verso il μC costituito da una PIC a cui è affidato il compito di estrarle dal formato NMEA, modificarle in modo da poterle dividere in gruppi di otto bit e inviarle al blocco Parallelizzazione in corrispondenza di un segnale di trigger. Quest'ultimo blocco, formato dalla PLD, ha il compito di concatenare i byte in cui è divisa

l'informazione, memorizzarla e renderla accessibile al bus su un opportuno comando di lettura.

Il segnale 1PPS è inviato al blocco Counter formato anche esso come il blocco Parallelizzazione da un PLD; quando arriva, i contatori implementati nelle PLD, iniziano il loro conteggio scandito dal segnale proveniente dall'oscillatore che costituisce il blocco Clock. Quando arriva un segnale di trigger il numero di conteggi viene memorizzato per renderlo poi disponibile in uscita quando è presente il comando di lettura.

In questo capitolo è descritta la programmazione della PIC e delle PLD. Nella realizzazione del programma della PIC si è inizialmente previsto un segnale di ingresso, che ha la funzione di segnalare un errore quando si verifica una cattiva trasmissione dei dati ed uno di uscita a cui è affidato il compito di resettare il sistema verso cui la PIC invia le informazioni.

A questo programma sono state apportate successivamente delle modifiche consistenti in una diversa gestione dei segnali e nell'aggiunta di alcune istruzioni che permettono l'invio di comandi al GPS.

L'uso della PIC è risultato necessario poiché le operazioni che essa compie risultano impossibili da implementare su una PLD, non adatta a svolgere operazioni eccessivamente complesse.

L'informazione che la PLD riceve dalla PIC è formata da gruppi di otto bit sulla linea dei dati (**tabella 4**) ognuno dei quali è seguito sulla linea di controllo dal corrispettivo codice identificativo e da un segnale indicante che il dato è pronto per essere acquisito ed utilizzato come clock. L'idea di partenza su cui si è sviluppato l'algoritmo e la successiva implementazione, consiste nell'identificare i diversi gruppi di byte contenenti le informazioni ricevute, discriminandoli in base al loro opcode e al numero di byte in cui sono contenuti.

DATO	NUMERO DI BYTE
UTC	4
Data	3
Latitudine	4
Longitudine	4
Altitudine	2
Numero di satelliti	1
DOP	3

Tabella 4 Ordine di trasmissione dei dati

Anche nella programmazione delle PLD come in quella della PIC si sono esaminate due alternative; in questo caso, le due strade seguite non differiscono solo nel programma implementato su di esse, ma anche nel dispositivo impiegato. I modelli di PLD utilizzati sono stati inizialmente le XC9572-PC44-7 e successivamente le Spartan XCS10-PC84. La prima è

composto da 44 piedini, di cui 34 possono essere utilizzate come linee di I/O, dei rimanenti alcuni sono di alimentazione e altri riservati alla programmazione; la seconda è composta da 84 piedini di cui 61 sono usati come linee di I/O. La scelta di questo secondo modello è dovuta al fatto che esso, grazie al maggior numero di pin di I/O disponibili, consente di utilizzare una sola PLD per ogni informazione ricevuta. Infatti, poiché il numero di bit di cui un'informazione necessita per essere visualizzata può arrivare ad un massimo di 32 come nel caso dell'UTC e dovendo prevedere altre linee di I/O per la gestione di altri ingressi o uscite, occorrono due XC9572 poiché 34 linee di I/O non sono sufficienti.

Nella prima parte esamineremo la programmazione delle XC9572 e la costruzione del relativo circuito di test, fornendo alcuni spunti futuri per una ulteriore implementazione dei programmi sviluppati; successivamente vengono sviluppati i programmi per l'implementazione del programma relativo agli XCS10 e viene costruito un circuito stampato per la loro verifica. La programmazione di una PLD può essere eseguita tramite due linguaggi equivalenti, VHDL e Schematic; in questo contesto viene utilizzato quest'ultimo per l'ampia visibilità che esso dà al codice inserito, essendo un linguaggio di programmazione grafico.

Il programma che viene implementato è costituito da diversi blocchi, ciascuno dei quali ha una particolare funzione: essi possono essere delle porte logiche, dei buffer di ingresso o uscita, flip-flop e altro.

Il compilatore utilizzato per programmare la PLD è il PLD ISE 7.1i che fornisce un utile strumento di verifica del programma immesso, simulando i risultati che esso deve fornire.

3.2 Programmazione della PIC

Come già detto in precedenza, le informazioni inviate dal GPS sono contenute in una serie di stringhe in formato NMEA; le stringhe GSA, RMC, GGA, forniscono le informazioni di interesse: nella prima sono contenute l'UTC, la latitudine, la longitudine, l'altitudine e il numero di satelliti; nella seconda il giorno, il mese e l'anno e nella terza il PDOP, l'HDOP, il VDOP indicanti la disposizione che hanno i satelliti nello spazio.

Il dispositivo utilizzato per acquisire queste stringhe ed estrarne le informazioni in esse contenute è costituito da una PIC.

La PIC legge continuamente le informazioni prodotte dal GPS, fino a quando non interviene il segnale di trigger dal rivelatore di particelle. La PIC, inizia quindi l'esecuzione di una nuova parte di programma che consente, dopo aver eseguito delle opportune elaborazioni, di inviare tali informazioni in gruppi di otto bit insieme ad un codice identificativo (opcode) verso le PLD, al fine di verificare la corretta lettura delle informazioni.

A tale scopo, è stato previsto inoltre un display LCD, collegato alla PIC. L'utilizzo del codice identificativo è stato reso necessario per introdurre un

metodo di intercettazione degli errori che possono verificarsi nella trasmissione e ricezione delle informazioni e per far riconoscere alla PLD quale informazione essa sta ricevendo.

L'opcode è formato da 5 bit, poiché vengono inviati 21 byte. La **Tabella 5** riportata qui di seguito contiene la lista dei byte inviati con il rispettivo codice identificativo.

Tabella 5 Opcode

N°	DATO	OPCODE
1	Otto bit più significati dell'UTC	00001
2	Secondi otto bit dell'UTC	00010
3	Terzi otto bit dell'UTC	00011
4	Otto bit meno significativi dell'UTC	00100
5	Giorno	00101
6	Mese	00110
7	Anno	00111
8	Otto bit più significativi della latitudine	01000
9	Secondi otto bit della latitudine	01001
10	Terzi otto bit della latitudine	01010
11	Otto bit meno significativi della latitudine	01011
12	Otto bit più significativi della longitudine	01100
13	Secondi otto bit della longitudine	01101
14	Terzi otto bit della longitudine	01110
15	Otto bit meno significativi della longitudine	01111
16	Otto bit più significativi dell'altitudine	10000
17	Otto bit meno significativi dell'altitudine	10001
18	Numero di satelliti rilevati	10010
19	PDOP	10011
20	HDOP	10100
21	VDOP	10101

Esaminiamo le istruzioni che la PIC esegue, nel caso in cui non vi sia nessun segnale di trigger. Esse sono:

1. **Inializzazione dell'interrupt:** viene attivato l'interrupt proveniente dalla linea di I/O della PIC identificata con RB0.
2. **Inializzazione delle linee di I/O della PIC:** vengono definiti gli ingressi della PIC e le loro modalità di utilizzo.
3. **Verifica del corretto funzionamento dei dispositivi di uscita:** viene visualizzato sul display la scritta "DATI PROVENIENTI DAL GPS".
4. **Controllo del livello logico assunto dalla linea d'ingresso relativa all'errore:** quest'ingresso, collegato ad una uscita della PLD,

raggiunge un livello alto se si verifica un interrupt e una cattiva ricezione dei dati, che devono essere rinviati dalla PIC alla PLD.

5. **Inizializzazione della variabile CONTROLLO:** viene posta a 0 la variabile, che serve per il controllo dell' esatto valore dell'UTC effettuato durante l'esecuzione delle istruzioni relative all' interrupt.
6. **Estrazione dei dati ricevuti serialmente:** fra le stringhe ricevute, contenenti i dati inviati dal GPS in formato NMEA, vengono intercettate quelle contenenti le informazioni di interesse, e da esse vengono estratte.
7. **Inizializzazione delle variabili SECONDI2 e GIORNO2:** servono per il controllo dell'errore che si può verificare nell'UTC e nella data. Successivamente la variabile CONTROLLO viene portata a 1 sempre per l'esecuzione di tale controllo.
8. **Verifica dell'esattezza delle istruzioni:** si esegue una verifica dei dati intercettati controllando il numero di satelliti visibili. Le informazioni ricevute vengono giudicate attendibili se il numero dei satelliti è compreso fra 4 e 12. Il controllo sul numero dei satelliti è verificato se questo è superiore a tre, quindi si ha la visualizzazione delle informazioni. Il minimo numero di satelliti scelto è quattro poiché per una misura tridimensionale abbiamo bisogno di almeno quattro satelliti affinché essa sia corretta (latitudine, longitudine, altitudine, ed uno come riferimento temporale). La scelta di un massimo numero uguale a 12 invece, è dovuto al fatto che il ricevitore non può agganciare un maggior numero di satelliti. Può succedere che tale numero sia zero, in questo caso però il valore che assume la variabile utilizzata nel programma sarà quello relativo al checksum, superiore sicuramente a 12. Se il range di valori che il numero di satelliti può assumere non fosse limitato superiormente si rischierebbero di giudicare attendibili informazioni prive di significato.
9. **Visualizzazione dei dati sul display LCD:** se i dati risultano attendibili sul display vengono visualizzati in successione i dati intercettati ed estratti, cioè:
 - UTC.
 - Data.
 - Latitudine.
 - Longitudine.
 - Altitudine.
 - Numero di satelliti.
 - VDOP, HDOP, PDOP.

Se le informazioni non risultano attendibili sul display appare la seguente stringa: "NUMERO DI SATELLITI INSUFFICIENTE".

Eseguite tutte le istruzioni sopra riportate il programma le ripete a partire dal punto tre. Questo ciclo si ripete in maniera continuata fino a quando non interviene il segnale di trigger. In tal caso il programma sospende l'istruzione in esecuzione per eseguire quelle relative alla gestione dell'interrupt. Esse sono:

1. **Elaborazione dei dati estratti:** in questa fase avviene la trasformazione dei dati estratti e successivamente la loro suddivisione in gruppi di otto bit per essere trasmessi. I dati che necessitano di essere manipolati per essere più facilmente ricostruibili dall'apparato ricevitore sono UTC, latitudine e longitudine, HDOP, VDOP e PDOP. I dati riguardanti la latitudine, e la longitudine, prima di essere inviati, vengono trasformati in millesimi di minuto; ciascuno dei valori ottenuti è trasformato in binario e diviso in due parti, immagazzinate in due variabili word (16 bit): una contenente i bit più significativi, l'altra quelli meno significativi. Il bit più alto della variabile contenente i bit più significativi, viene usato per contenere l'informazione riguardante la coordinata di latitudine o longitudine. Questa scelta è stata dettata dal fatto che il valore assunto dalla latitudine o dalla longitudine non potrà mai essere talmente elevato da poter portare a 1 questo bit. In particolare, esso è 1 se la coordinata di latitudine è N (E per la longitudine), 0 se è S (W per la longitudine).

L'altro dato che necessita di elaborazione è quello contenente l'informazione sull'UTC. Di esso, prima di eseguire qualsiasi operazione, viene controllata l'esattezza. Tale controllo, necessario a causa della bassa velocità di lettura della porta seriale, la quale può far sì che l'interrupt possa avvenire prima che l'UTC venga aggiornato al valore corretto, consiste nel verificare il valore che assumono le variabili CONTROLLO e SECONDI2. In particolare, se subito dopo l'inizio di una nuova lettura dal GPS interviene l'interrupt, il programma, prima di iniziare la procedura di gestione dei dati riguardanti l'orario UTC controlla se il valore della variabile CONTROLLO sia zero o uno; se è zero, e SECONDI è uguale a SECONDI2, significa che è iniziato un nuovo ciclo di lettura dei dati ma il valore dell'UTC non è stato ancora aggiornato, risultando quindi inferiore a quello esatto di un secondo. Sarà il programma ad incrementarlo al fine di fargli assumere l'esatto valore.

Dopo aver effettuato tale controllo il programma trasforma le ore, i minuti e i secondi che compongono il valore dell'UTC inviato dal GPS in secondi (in realtà il GPS invia i dati riguardanti l'UTC fino ai millesimi di secondo ma essi vengono ignorati: la parte inferiore ai secondi infatti viene presa in considerazione dalle PLD che raggiunge la precisione dei nanosecondi). Anche il valore dell'UTC, come già fatto

per latitudine e longitudine, viene diviso in bit più significativi e bit meno significativi e immagazzinato in due variabili word.

Un controllo analogo a quello eseguito sull'UTC viene eseguito per verificare l'esatto valore del giorno: esso effettua la correzione della variabile corrispondente, se CONTROLLO è uguale a zero, se GIORNO2 è uguale a GIORNO e se le variabili, in cui è immagazzinato il tempo sono poste a zero. Se queste condizioni si verificano, significa che è iniziato un nuovo giorno, (infatti solo in quel caso le variabili riguardanti l'UTC assumono tale valore, questo perché l'informazione che lo riguarda, inviata dal GPS è 000000 00 ore, 00 minuti, 00 secondi), e il valore della variabile GIORNO non è stato aggiornato, risultando inferiore di un'unità a quello corretto. Anche in questo caso il programma correggerà tale valore incrementando la variabile.

I valori di PDOP, HDOP, VDOP, essendo in numero decimale, vengono trasformati in interi e inviati.

2. **Invio dei dati:** tutti i dati acquisiti serialmente, sia quelli che hanno avuto bisogno di essere modificati, sia quelli che non necessitavano di tale modifica sono divisi in gruppi di otto bit e immagazzinati in un vettore, pronti per essere inviati verso le PLD. La PIC, prima di effettuare quest'operazione, manda un segnale di reset alla PLD e, successivamente, trasmette i dati nel seguente ordine:

- Dato.
- Opcode che lo identifica.
- Clock il cui scopo è quello di segnalare alla PLD che l'informazione è pronta per essere acquisita.

Il programma dopo aver inviato tutti i dati contenuti nel vettore, non appena rientrato nella sua esecuzione normale, invia un secondo segnale di reset per cancellare i valori presenti sulle uscite della PLD.

Prima della gestione dell'interrupt viene eseguito un controllo sul numero dei satelliti identico a quello eseguito nell'esecuzione normale del programma. Le istruzioni sopra descritte vengono eseguite solo se l'esito di questo controllo risulta essere positivo.

Nel programma non si è mai fatto uso della memoria EEPROM di cui la PIC dispone per l'immagazzinamento delle informazioni che essa riceve. Ciò è dovuto al fatto che un processo di lettura e scrittura su l'EEPROM necessita di un tempo molto lungo, risorsa che in questo particolare ambito è di fondamentale importanza. Infatti, tutte le scelte riguardanti sia la fase di programmazione sia quella di scelta dei componenti, sono fatte al fine di ottimizzare il tempo. Questo spiega perché le informazioni non vengono mai conservate dalla PIC, ma solo lette, visualizzate e, nel caso di richiesta, inviate verso unità esterne.

3.3 Test del programma

Il prototipo realizzato per il test è mostrato in **figura 11**, con il relativo schema circuitale in **schema 1**.

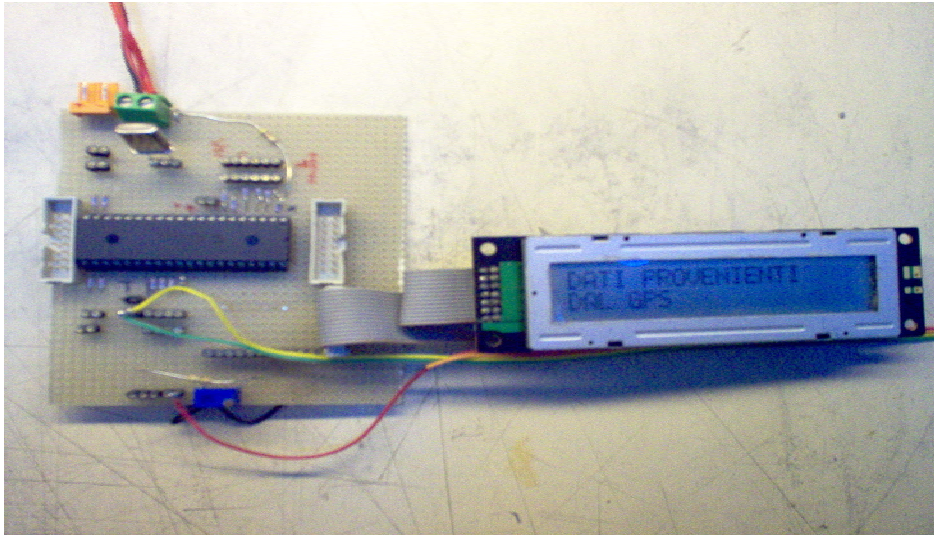
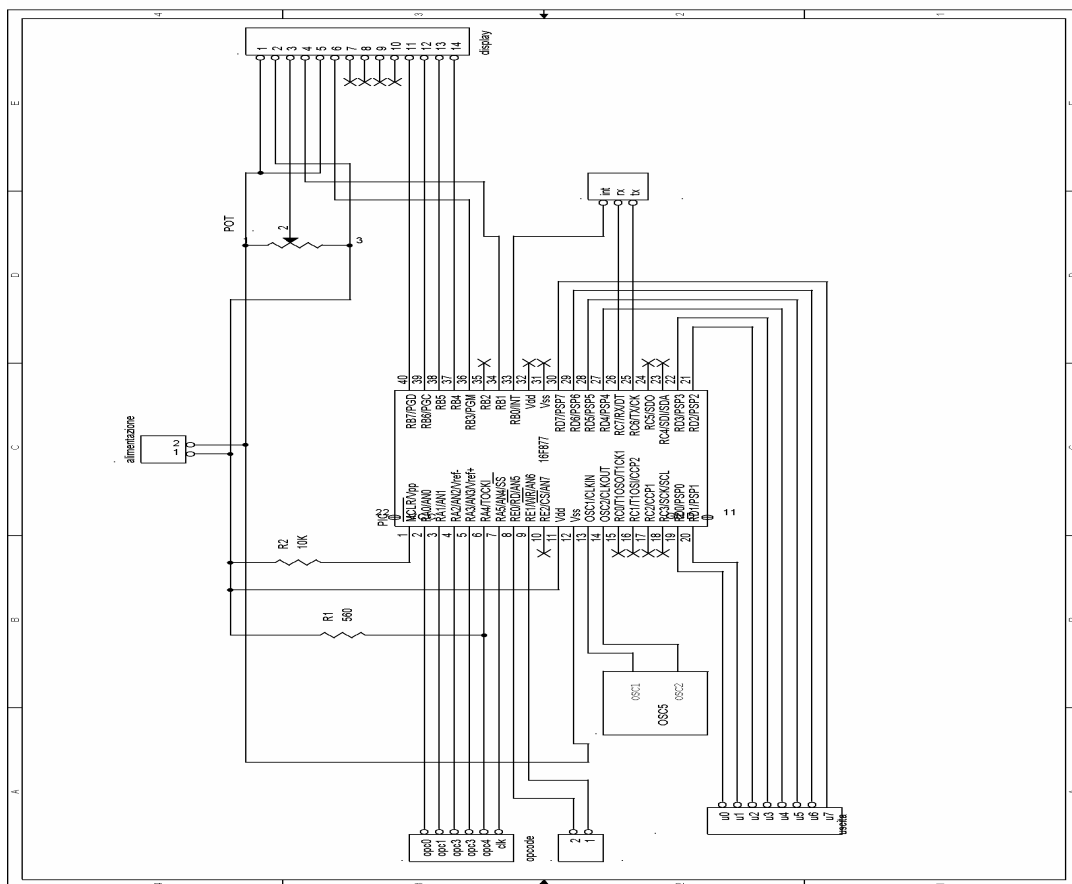


Figura 11 Circuito di test.



Schema 1

Per ottimizzare i tempi di sviluppo e test si sono adoperati sei connettori di trasmissione dati connessi con le sonde dell'analizzatore di stati logici e con i connettori della PLD.

Nel nostro caso, come si può vedere nello **schema 1**, le porte della PIC sono ripartite nel seguente modo:

- La porta B è stata utilizzata per gestire le informazioni necessarie al funzionamento del display LCD, fatta eccezione per RB0, il quale è utilizzato come ingresso per l'interrupt.
- Tramite la porta A, si inviano le informazioni riguardanti l'opcode (da RA0 a RA4) e il clock (RA5).
- Tramite la porta E si invia il segnale di reset (RE1) e si riceve il segnale di errore (RE2)
- Tramite la porta D si inviano i dati in uscita verso le PLD.

Queste porte possono essere utilizzate sia per lo scambio d'informazioni in formato analogico sia in formato digitale. Per utilizzarle in quest'ultimo modo, si è fatto uso all'interno del programma dell'istruzione $ADCON1 = 7$, necessaria per definire queste porte di tipo digitale.

In **figura 12** è rappresentata la sequenza di invio dei dati come che appare sull'analizzatore, quando è collegato al circuito e interviene l'interrupt, generato collegando RB0 a Vcc.

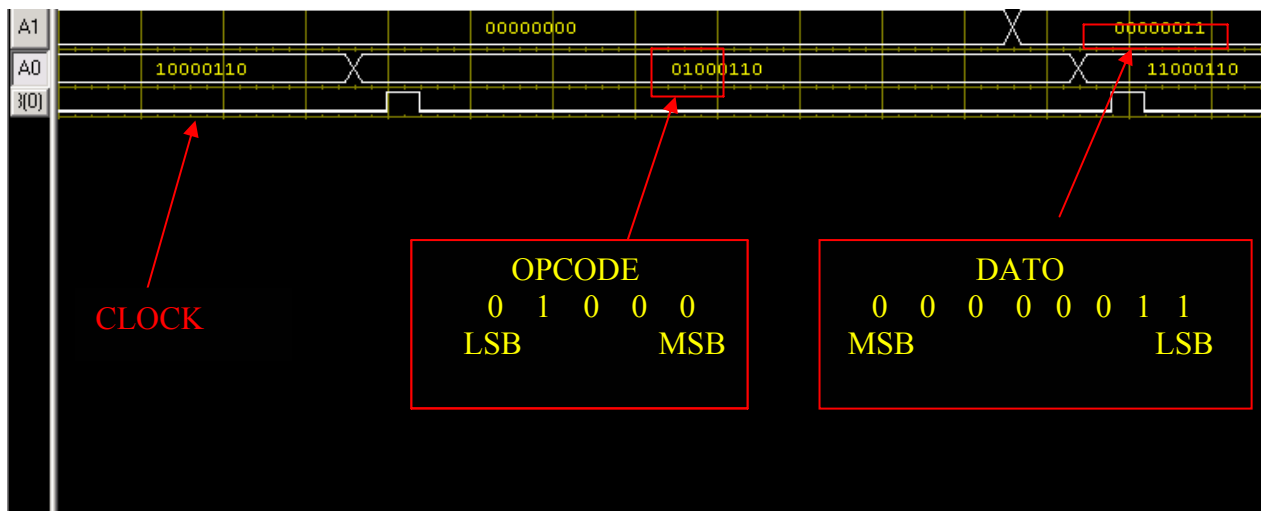


Figura 12 Dati visualizzati sull' Analizzatore di stati logici

Il canale su cui avviene la prima transizione è A1 seguito da A0 e dal clock. Le transizioni dell'opcode sono 10000 - 01000 - 11000, ciò significa che il dato visualizzato riguarda i tre byte più significativi dell'UTC. Su A1 non vi è cambiamento corrispondente alla prima variazione dell'opcode poiché i primi

due byte dell'UTC sono 0, ciò significa che l'orario ricevuto è inferiore a 65536 secondi (18^h 12^m 26^s).

L'interfacciamento seriale col GPS avviene attraverso la porta RC

In **figura 13** è riportato il circuito sul quale è montato il ricevitore GPS.

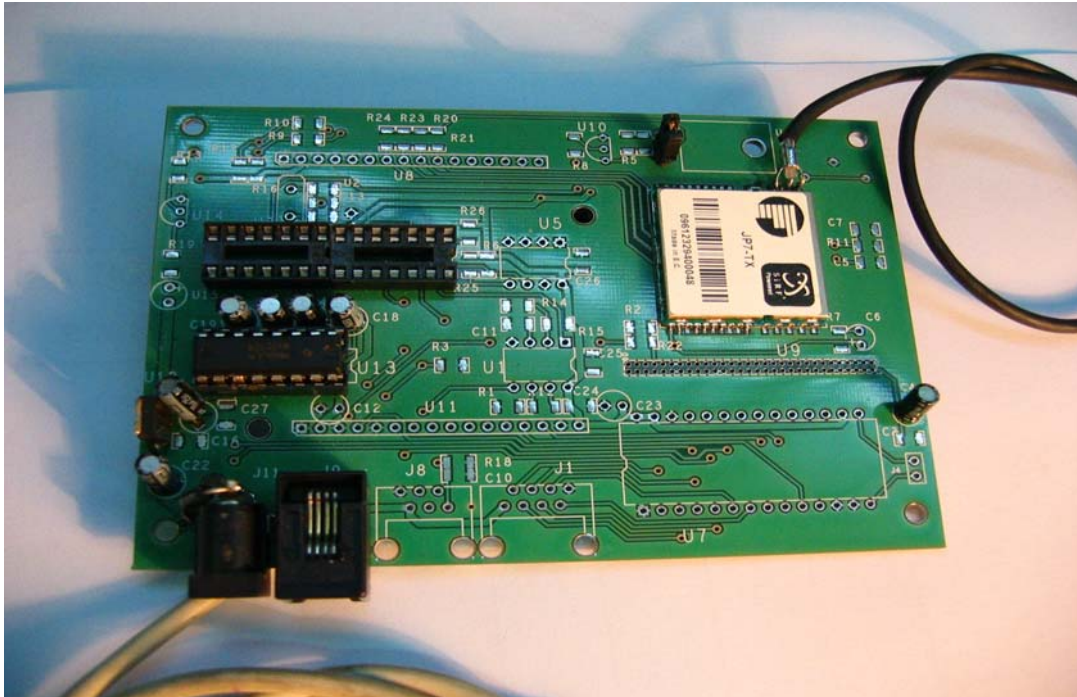


Figura 13 Falcom JP7

Questo ricevitore è il FALCOM JP7, che viene utilizzato per poter testare i programmi di estrazione dei dati ricevuti. Esso non verrà utilizzato nella fase finale del progetto poiché non prevede il segnale PPS. Sarà dunque utilizzato un altro tipo di ricevitore, il Resolution T della Trimble che, al contrario, lo fornisce, garantendo un'incertezza dell'ordine di ± 15 ns, che consente misure estremamente precise.

3.4 Sviluppo successivo del programma

Come detto ad inizio capitolo, il programma riguardante la PIC ha subito nello svolgimento della tesi alcune modifiche, riguardanti i segnali che essa deve gestire. In particolare:

- Non vi è nessun segnale di reset implementato necessario alla cancellazione dei dati immagazzinati nelle PLD poiché tale segnale è affidato all'unità esterna verso la quale le PLD devono trasmettere i dati, esso viene inviato dopo che quest'unità legge tutte le informazioni di cui necessita

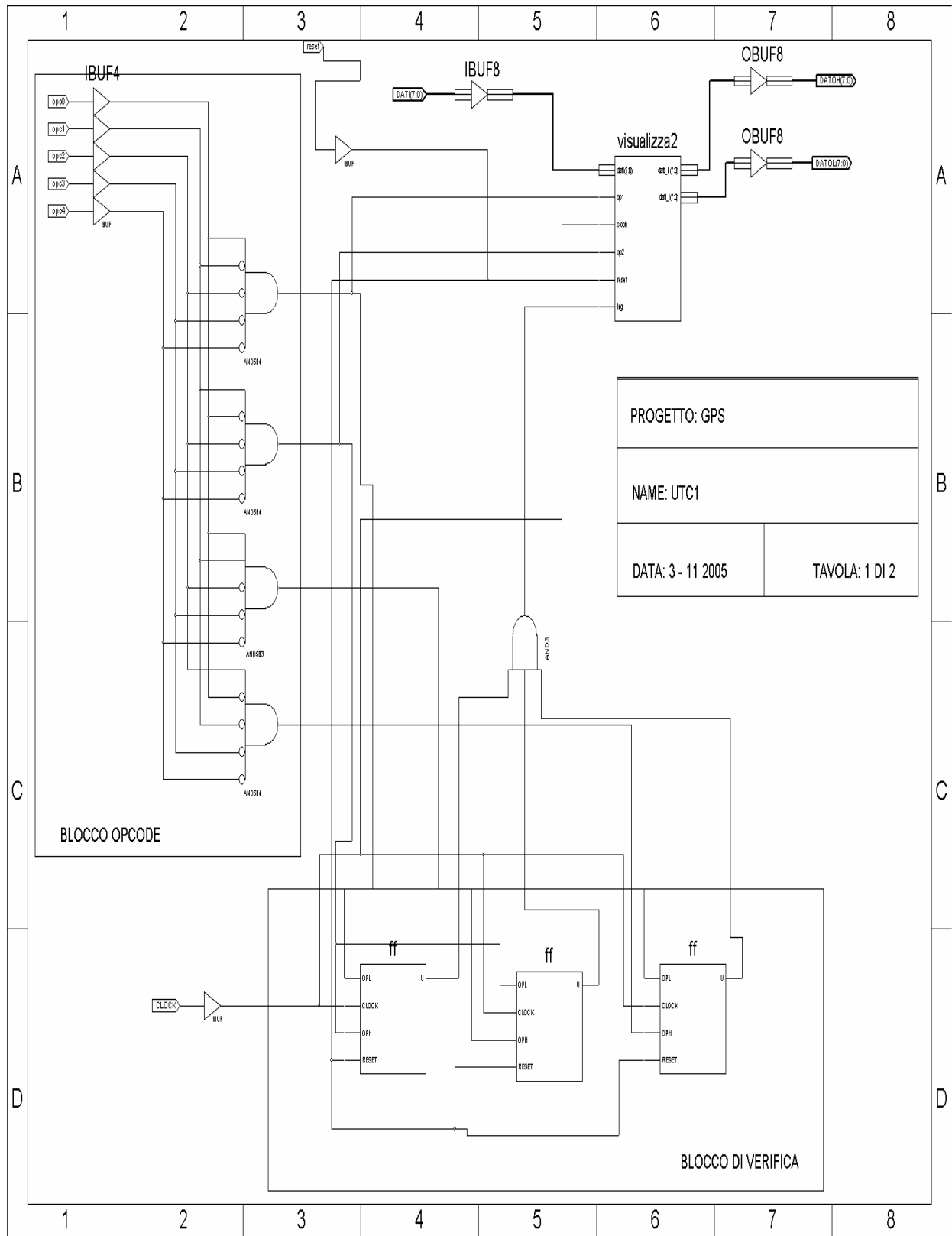
- È stato eliminato il segnale di errore poiché le PLD e la PIC sono contenute nello stesso circuito stampato e la possibilità che si verifichi un errore di trasmissione fra queste due unità è pressoché nulla.
- È previsto un segnale in ingresso alla PIC, proveniente dall'unità esterna, il cui compito è quello di abilitare l'invio dei dati se interviene un segnale di trigger. Questo segnale abilita o impedisce tale invio, a seconda se l'unità ha completato o no la lettura dei dati presenti in uscita dalle PLD. Se quando interviene l'interrupt tale segnale è attivo (cioè ha un livello logico alto), la PIC attende che il suo livello logico sia basso, prima di inviare i dati alle PLD.

3.5 Sviluppo del programma per le XC9572-PC44

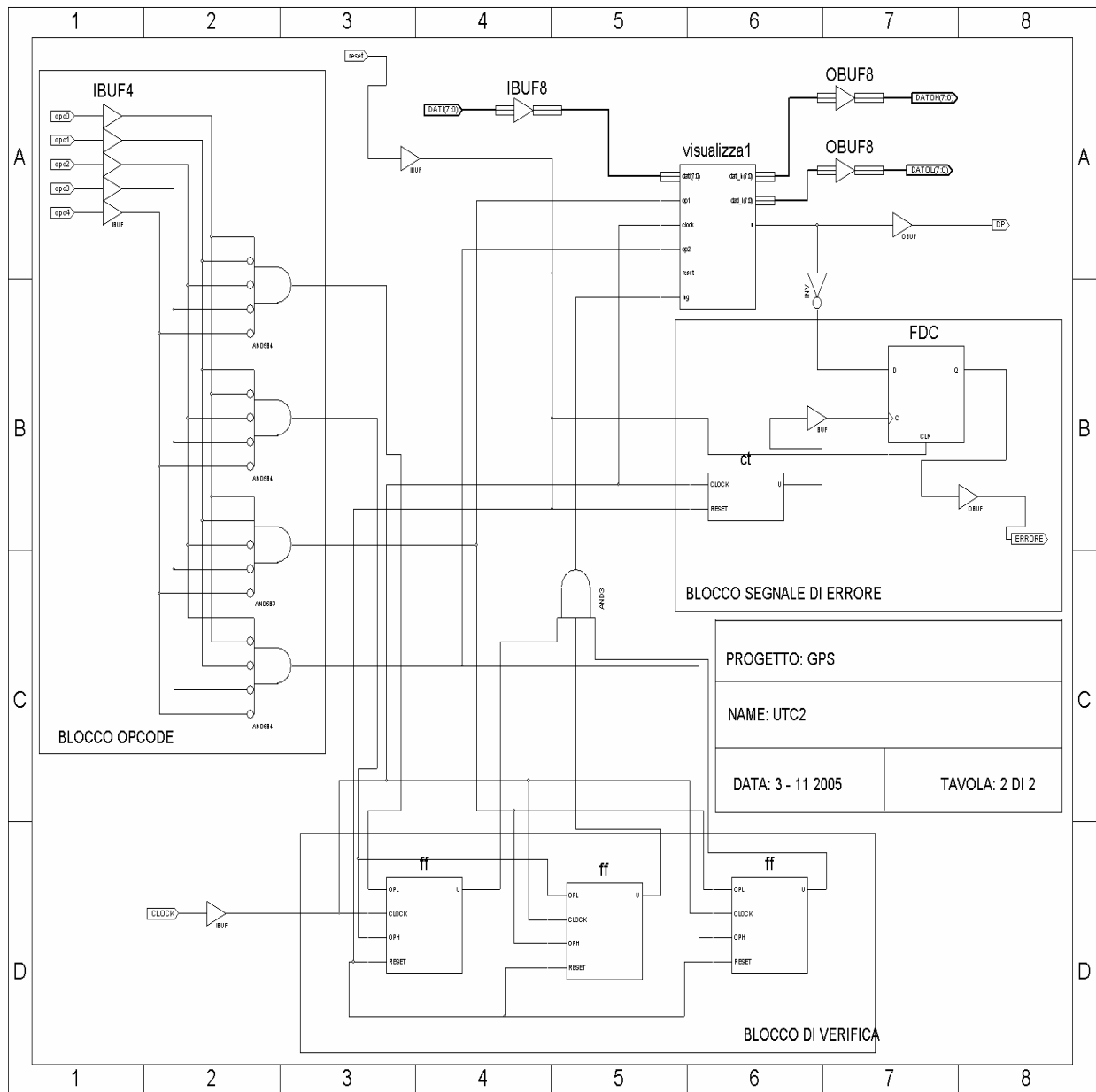
Al fine di conoscere i dati riguardanti il numero di satelliti e l'altitudine, essendo necessari solo rispettivamente 8 bit e 16 bit, per ognuno di essi viene utilizzata una sola PLD. Per tutti gli altri dati, essendo necessari da un minimo di 24 bit a un massimo di 32 bit ne vengono utilizzate due.

Il programma implementato sulle PLD montate sul circuito di test è quello relativo alla ricezione dell'UTC. Fra tutte le informazioni che riceve, la PLD ricava quella relativa a esso, discriminandola fra tutte le altre in base agli opcode che lo identificano (00001, 00010, 00011, 00100). I programmi implementati sulle due PLD sono riportati negli **schemi 2 e 3** seguenti.

Nei programmi ci sono due macro blocchi comuni il BLOCCO OPCODE e il BLOCCO DI VERIFICA. Il primo ha la funzione di riconoscere gli opcode che sono associati alle informazioni riguardanti l'UTC ed è composto da una serie di porte AND, di cui alcuni ingressi sono negati in maniera opportuna, affinché quando si presentano gli opcode di interesse almeno una di tali porte abbia in uscita 1 logico. Le porte sono disposte ordinatamente, in modo che la prima faccia riferimento al primo opcode, la seconda al secondo e così via, seguendo l'ordine di invio impostato nel programma della PIC. Il secondo blocco ha la funzione di verificare che gli opcode siano ricevuti nel giusto ordine.



Schema 2

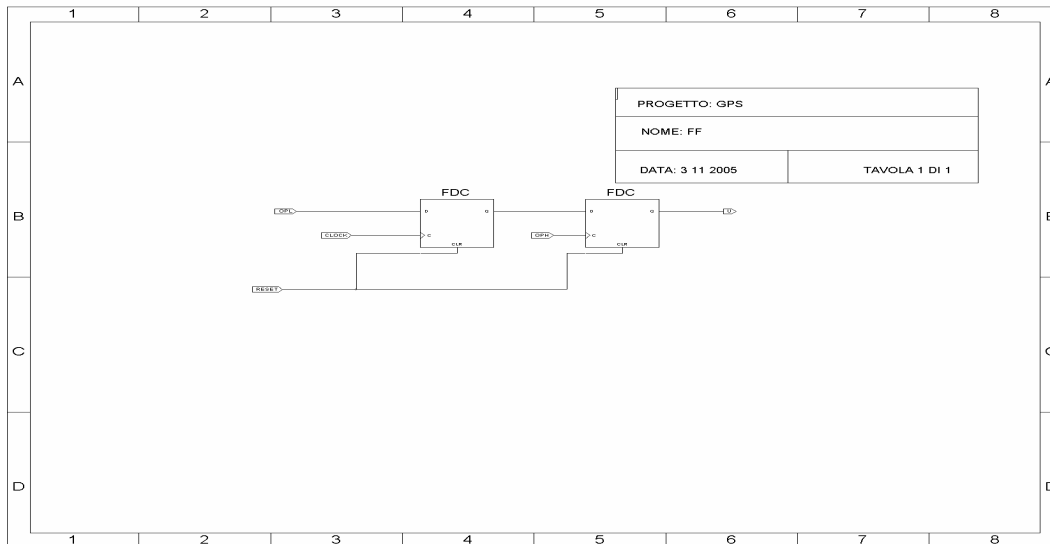


Schema 3

Questo blocco è costituito da alcuni simboli rappresentanti dei sottoprogrammi implementati nel programma principale per aumentare la visibilità di quest'ultimo.

Nello **schema 4** successivo è mostrato il programma contenuto in essi.

Esso è composto dalla serie di due flip – flop: il primo ha come ingresso il segnale “opl” corrispondente all’uscita di una delle porte AND del BLOCCO OP CODE, mentre il segnale di commutazione è dato dal clock proveniente dalla PIC. Il secondo ha per clock il segnale “oph”,



Schema 4

corrispondente all’uscita della porta AND immediatamente successiva. Viene utilizzata questa struttura piuttosto che un unico flip – flop con ingresso il segnale opl e clock oph per evitare errori dovuti alle transizioni di questi due segnali. Infatti, nello stesso istante in cui oph si porta alto, opl va a 0. Quindi il flip – flop, che è attivo sul fronte di salita del clock, trasferisce il segnale presente sul suo ingresso in uscita nel momento in cui commuta. Con la struttura utilizzata il segnale di arrivo del primo opcode viene immagazzinato grazie al clock che la PIC invia subito dopo di esso ed è quindi un dato stabile, quando arriva il secondo codice identificativo, pronto per essere inviato in uscita dall’ultimo flip – flop.

Questo procedimento è eseguito su tutte le transizioni dell’opcode (00001 → 00010, 00010 → 00011, 00011 → 00100), se questi giungono nell’ordine prestabilito, tutti gli ingressi della AND ai quali queste uscite sono collegate si portano a 1 e quindi quest’ultima porta il segnale corrispondente la sua uscita a 1.

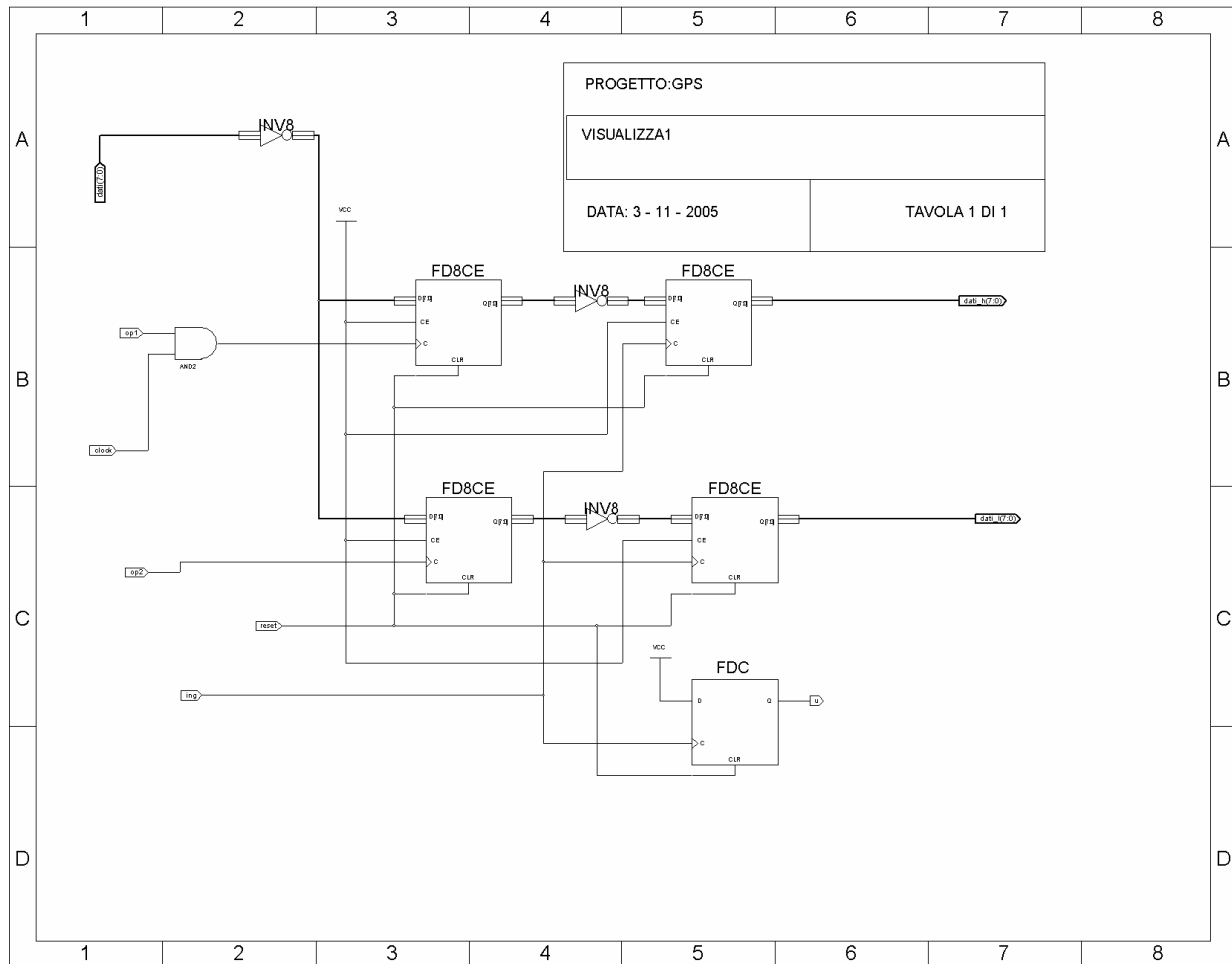
Questo segnale costituisce un ingresso del simbolo chiamato “visualizza2” o “visualizza1” a seconda se si considera lo schema relativo a UTC1 o UTC2.

Anche questi simboli come quello precedente sono dei sottoprogrammi. Quello riguardante visualizza1 è riportato nello **schema 5**.

Lo schema riguardante visualizza2 è del tutto analogo, fatta eccezione per l’assenza del flip – flop FDC utilizzato per generare il bit indicante che il dato è pronto.

Oltre al segnale proveniente dalla AND (chiamato “ing”), in ingresso vi sono anche il clock, i segnali “op1” e “op2” corrispondenti rispettivamente agli opcode 00011 e 00100 (00001 e 00010 per visualizza2) e il dato contenente gli otto bit dell’informazione.

Questo programma è formato da due serie di flip – flop: la prima è riservata alla ricezione dei dati riguardante l’opcode 00011 (00001 per visualizza2)



Schema 5

la seconda per l'opcode 00100 (00010 per visualizza2). Il primo flip – flop della prima serie trasferisce in uscita il dato di ingresso se op1 si porta a 1 e arriva il segnale di clock immediatamente successivo all'opcode corrispondente, (infatti così la AND a cui esso è collegato va ad 1 e siccome essa costituisce il clock per il flip - flop che ha come ingresso i dati, trasferirà questi ultimi sulla sua uscita).

Il primo flip-flop della seconda serie trasferisce in uscita il dato quando op2 si porta a 1 (in visualizza2, anche quest'ultimo è sensibile alla commutazione dell'uscita di una porta AND, che ha come ingressi op2 e il clock corrispondente).

Gli ultimi due flip – flop hanno la funzione di sincronizzare l'uscita delle informazioni, rendendole visibili verso l'esterno solo quando tutte quelle relative all'UTC vengono ricevute. Infatti questi ultimi due flip – flop trasferiscono contemporaneamente i dati in uscita quando il segnale ing commuta da basso ad alto.

Il flip – flop supplementare presente in visualizza1 ha l'ingresso costantemente alto e riporta il valore in uscita parallelamente agli altri flip -

flop (anche esso ha sul clock il segnale ing), per indicare che il dato in uscita è corretto e pronto per essere acquisito.

Un blocco, presente in UTC2 ma assente in UTC1 è il BLOCCO SEGNALAZIONE DI ERRORE. Esso è composto dal simbolo contatore4 e da un ulteriore flip – flop. Il contatore4 contiene un sottoprogramma costituito da un contatore (il cui incremento è dettato dalle transizioni del clock), programmato per dare in uscita un livello alto dopo quattro colpi di clock (infatti tutti i dati relativi all’UTC sopraggiungono con i primi 4 byte ricevuti). L’uscita di questo oggetto costituisce il segnale di clock del flip – flop posto in cascata il quale ha in ingresso il segnale di “dato pronto” negato. Quando il contatore porta l’uscita da 0 a 1, il flip – flop trasferisce l’ingresso in uscita e, se la sequenza degli opcode è corretta, in ingresso ha uno 0, poiché il segnale di dato pronto è stato già portato a 1. Se, invece, la sequenza degli opcode non è quella desiderata, il suo ingresso è 1, (poiché il flip – flop che ha per uscita il segnale di dato pronto ha sul clock un segnale che non può transitare mai da 0 a 1, ma è fisso a 0), e quindi in uscita abbiamo un 1 che è il segnale corrispondente al segnale di errore.

Tutti gli oggetti utilizzati hanno un ulteriore ingresso che è quello di reset, esso ha il compito di resettare tutto il sistema, quando la PIC genera il segnale previsto per questo scopo.

3.6 Simulazione al PC

In **figura 14** è riportata la simulazione del programma quando non interviene nessun errore sulla sequenza degli opcode.

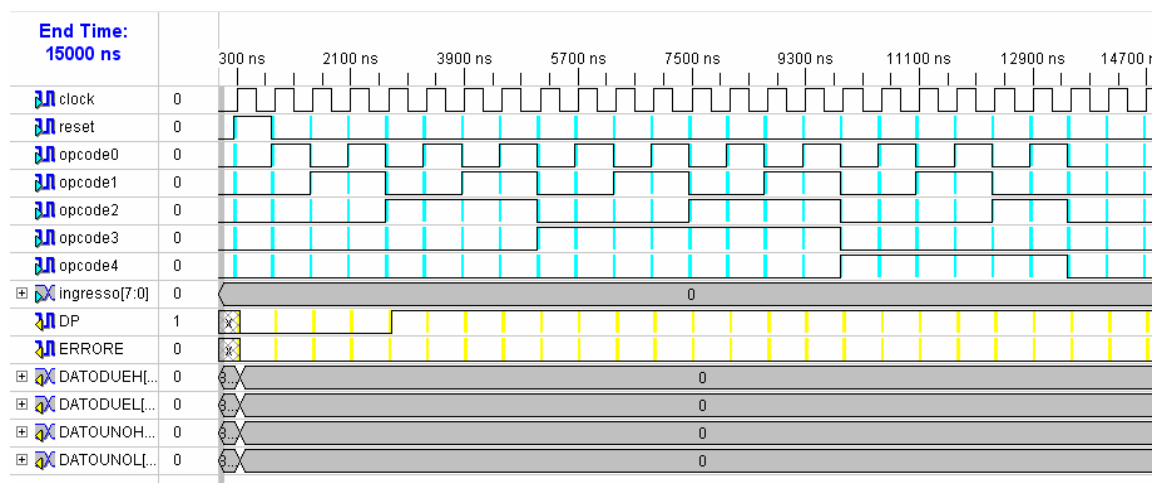


Figura 14 Simulazione del circuito senza la presenza di errori

Opcode0 indica il bit meno significativo, opcode4 quello più significativo. Come si può vedere, i dati riguardanti l’UTC vengono visualizzati dopo che

tutti i byte in cui è frazionato sono stati ricevuti, insieme al segnale DP di dato pronto.

In **figura 15**, è rappresentata una simulazione del comportamento del circuito, quando si verifica un errore.

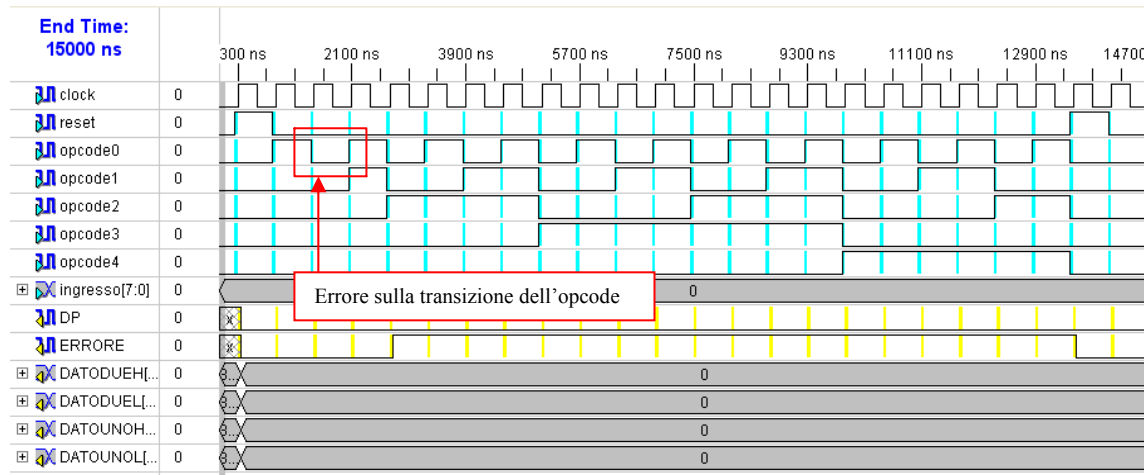
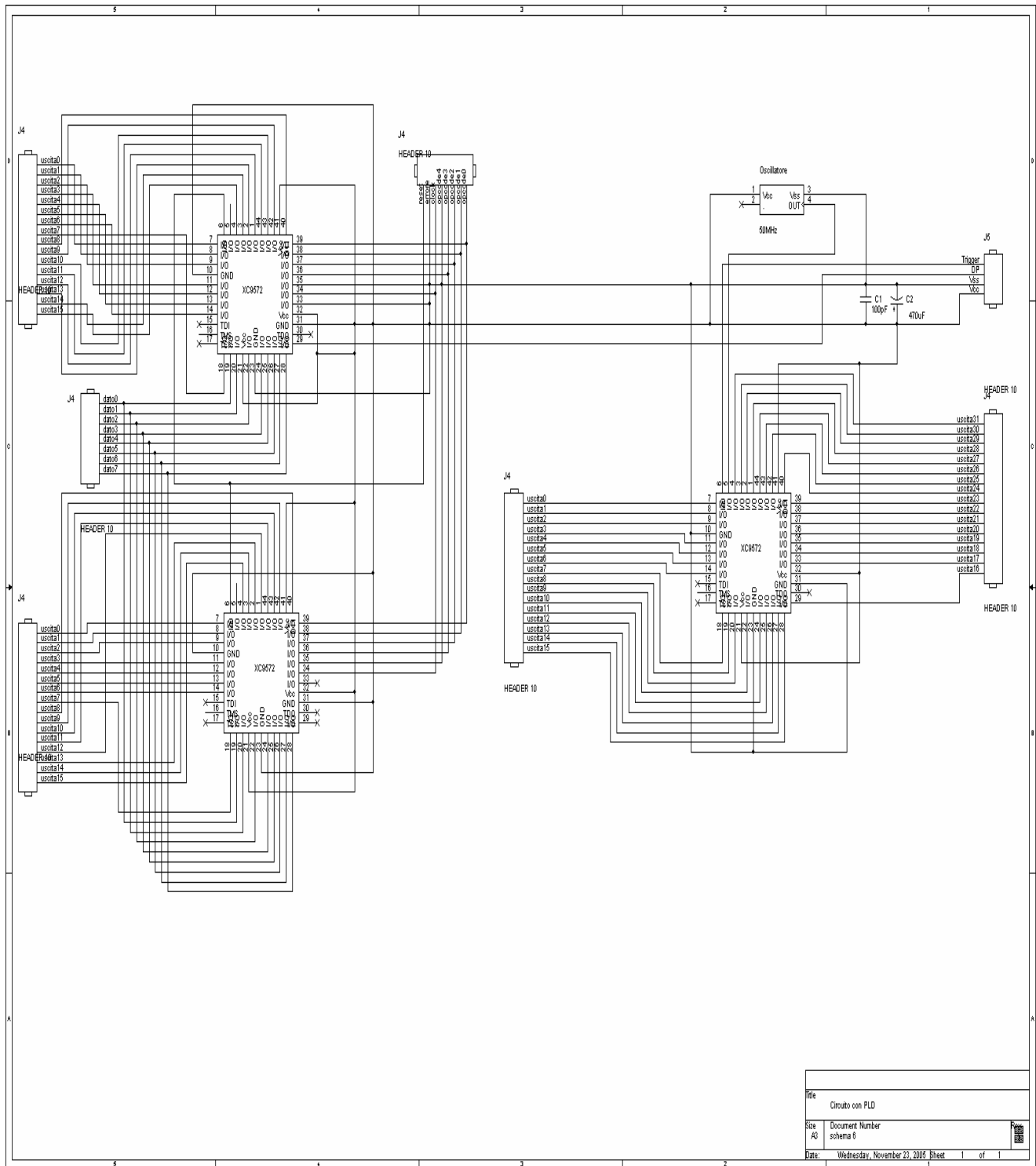


Figura 15 Simulazione del circuito con la presenza di errori

L'errore, in questo caso, interviene quando l'opcode, invece di passare dalla configurazione 00010 → 00011, passa da quella 00010 → 00000. Dopo essere stati effettuati quattro cicli di clock, necessari alla trasmissione dell'informazione relativa all'UTC, viene visualizzato il segnale di errore, mentre quello di dato pronto e quelli di uscita rimangono a 0.

3.7 Test del programma

Il prototipo realizzato è riportato nello schema 6 e in figura 16.



Schema 6

Nello schema ci sono tre PLD: quella più a destra viene utilizzata per la verifica del programma che deve fornire l'informazione relativa al tempo al di sotto del secondo.

Analizziamo la parte dello schema in cui è illustrata la parallelizzazione dei dati.

Sul circuito sono presenti diversi piedini, 32 di questi corrispondono ai dati parallelizzati, i quali vengono collegati, insieme ad un ulteriore piedino corrispondente al segnale di dato pronto, all'analizzatore di stati logici. Altri sedici sono collegati alla PIC, di cui otto servono per la ricezione dei dati, cinque per l'opcode, uno per il clock e gli ultimi due rispettivamente per il segnale di reset e di errore.

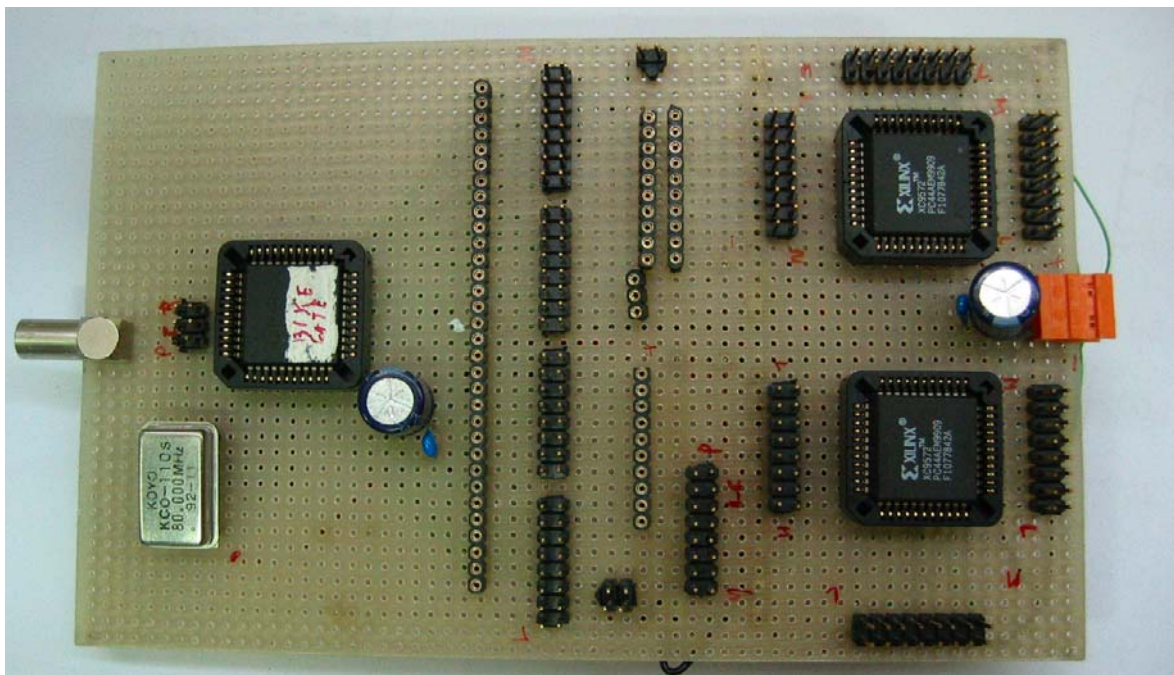


Figura 16 Circuito di test

Il risultato visualizzato sull'analizzatore di stati logici è riportato in **figura 17**.

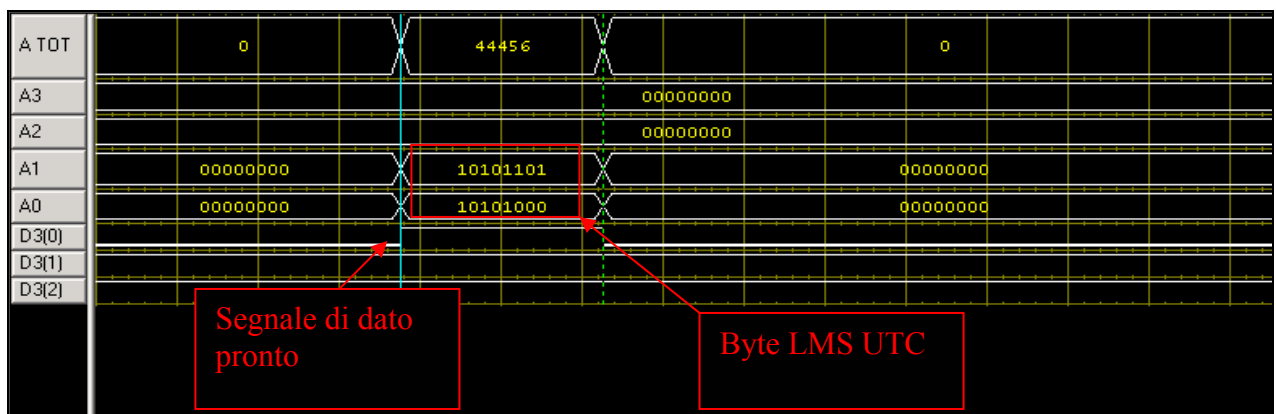


Figura 17 Dati visualizzati sull'analizzatore di stati logici

Sui canali A3 – A0 è visualizzata l'informazione relativa all' UTC e su D3(0) il segnale di dato pronto. Come si può vedere, inizialmente tutti i canali sono posti a zero, per via del reset che viene inviato alla PLD dalla PIC all'inizio della trasmissione. Dopo, viene effettuata la transizione sui vari canali. Sui canali A3, A2 non vi è transizione poiché l'orario visualizzato 12^h 20^m 56^s, una volta trasformato in secondi ha bisogno di 16 dei 32 bit disponibili per essere rappresentato, quindi i rimanenti (i più significativi), rappresentati sui canali A3 e A2, si trovano a zero.

Sul canale ATOT, viene visualizzato il valore in decimale ottenuto concatenando i valori presenti sugli altri canali di A. Esso serve per una più facile verifica se il dato visualizzato è corretto.

Alla fine tutti i canali si riportano a 0; ciò è dovuto al segnale di reset finale inviato dalla PIC.

3.8 Programma generale da realizzare per la ricezione delle informazioni

Il programma realizzato per la visualizzazione dell'UTC deve essere ripetuto, con le opportune modifiche, su tutte le PLD.

Le parti da modificare sono relative al BLOCCO OPCODE, infatti bisogna programmare gli ingressi delle porte AND di cui esso è costituito, al fine di rintracciare gli opcode associati all'informazione di interesse.

Un'altra modifica deve essere apportata al contatore, programmandolo in modo da portare la sua uscita alta, dopo aver raggiunto il valore dell'opcode associato all'ultimo byte di cui è costituita l'informazione per cui è programmata la PLD.

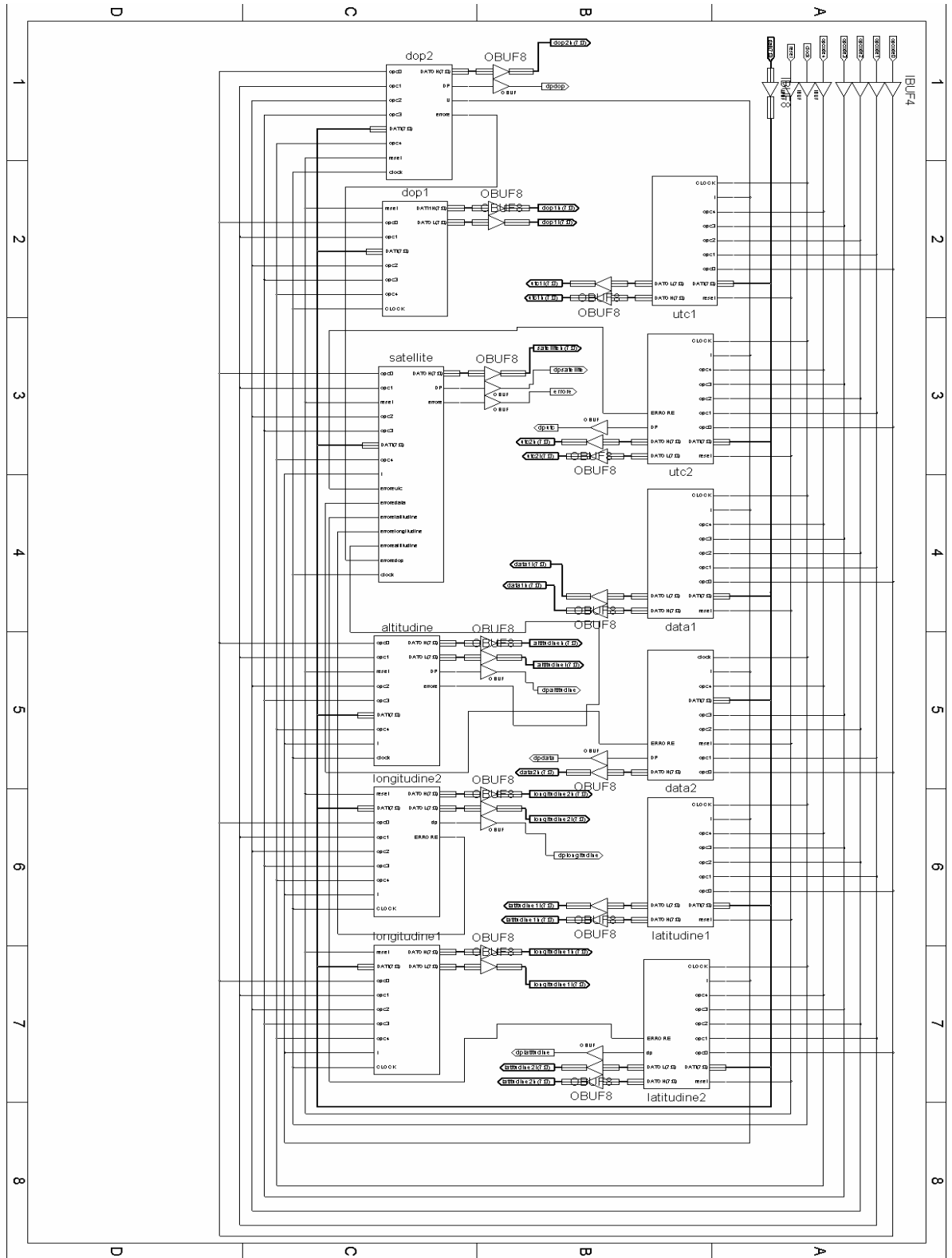
Infine, bisogna cercare di sincronizzare tutte le PLD, in modo tale che mandino in uscita le informazioni parallelamente, solo quando queste ultime sono tutte disponibili.

Per far ciò si è aggiunto ad ogni PLD un ulteriore segnale che costituisce un ingresso per tutte le PLD, fatta eccezione per quella programmata a ricevere l'ultima informazione inviata (nel nostro caso quella relativa al DOP) che invece genera il segnale. Questo segnale assume un livello logico alto solo quando la porta AND in cui confluiscono le uscite del blocco di verifica della PLD riservata alla ricezione del byte meno significativo del DOP, si porta a 1. Ciò significa che tutti gli opcode relativi all'informazione sul DOP sono giunti in maniera corretta, e quindi anche l'ultimo opcode è giunto a destinazione e tutta l'informazione è stata ricevuta.

Questo segnale viene inviato nel blocco visualizza di tutte le altre PLD, al quale è stato aggiunto un ulteriore ingresso. Questo blocco è stato modificato, rispetto a quello analizzato in precedenza, aggiungendo a valle di ciascuna delle due serie di flip – flop che lo costituiscono un altro flip - flop, il cui clock è costituito proprio dal nuovo segnale. Quando esso commuta dal livello basso a quello alto, se non ci sono stati errori nella ricezione, tutti i

flip-flop aggiunti nelle PLD hanno sulle loro linee d'ingresso il dato costituente l'informazione che viene visualizzato in uscita.

Lo **schema 7** è utilizzato per simulare il risultato finale che si vuole ottenere, una volta programmate tutte le PLD e collegate fra di loro.



Schema 7

Ogni singolo blocco rappresenta una PLD nella quale è stato inserito il programma che serve a visualizzare in parallelo i bit contenenti le informazioni. Ad ognuno di questi blocchi sono collegati otto linee di ingresso comuni a tutti. Sette di queste sono formate da un bit ciascuna: cinque servono per la comunicazione dell'opcode, una per la comunicazione del segnale di clock e l'altra per il segnale di reset. La linea d'ingresso rimanente è un bus di otto bit che serve per la comunicazione dei dati.

Ogni blocco ha un'uscita formata da uno o due bus di otto bit, a seconda del numero di bit necessari a contenere l'informazione, il cui ruolo è quello di visualizzare l'uscita delle informazioni in parallelo. I blocchi utilizzati sono:

- utc1 e utc2 per la visualizzazione dell'UTC.
- data1 e data2 per la data.
- latitudine1 e latitudine2 per la latitudine.
- longitudine1 e longitudine1 per la latitudine.
- altitudine per l'altitudine.
- satellite per il numero di satelliti.
- dop1 e dop2 per il DOP.

La gestione dell'errore, in questo caso, viene realizzata inviando il segnale di errore di tutte le PLD verso una che non necessitando di 16 bit per visualizzare il dato riguardante l'informazione da rappresentare, dispone di un numero sufficiente di piedini liberi che possono essere utilizzati come ingressi.

In questo caso la PLD in questione è quella riguardante i satelliti. Tutti questi segnali confluiscono in una porta OR che è stata implementata all'interno di questa PLD. In questo modo, quando almeno uno di questi segnali è a livello logico alto in uscita è presente un segnale di errore.

3.9 Simulazione al PC

Nella **figura 18** è riportata una simulazione del comportamento dell'intero circuito quando vi è una ricezione di informazioni senza che si verifichi nessun errore di ricezione di esse.

Come si può notare tutte le uscite vengono poste inizialmente a zero tramite il primo segnale di reset inviato dalla PIC, successivamente esse visualizzano il dato parallelamente. Solo il segnale di dato pronto viene associato alla ricezione completa di ogni informazione; dopo di che, tutte le uscite vengono riportate a zero tramite il secondo segnale di reset.

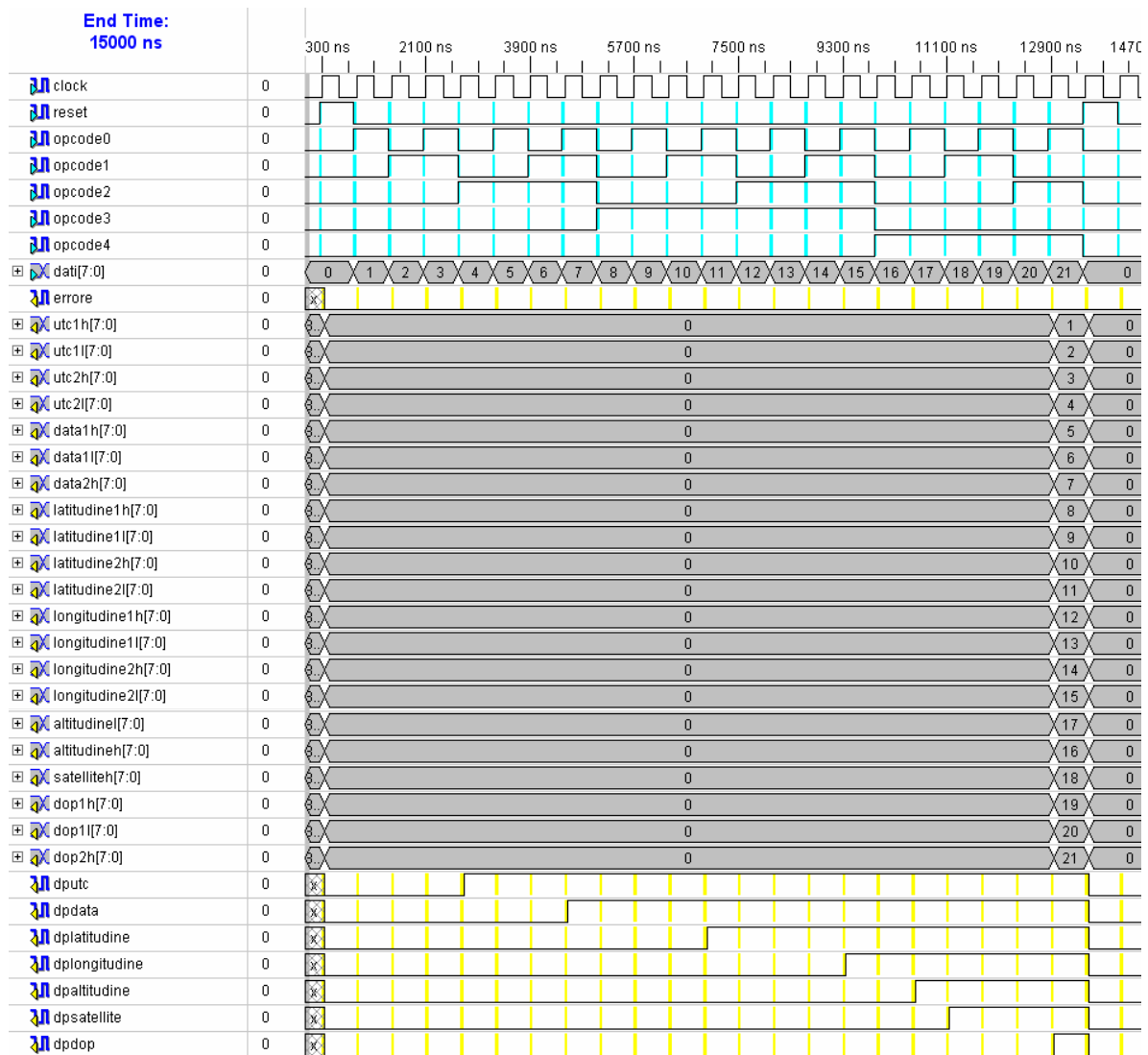


Figura 18 Simulazione del circuito senza la presenza di errori

La **figura 19** rappresenta la simulazione del comportamento del circuito quando vi è un errore nella ricezione.

In questo caso l'errore si verifica durante la ricezione della latitudine. Infatti, il segnale di dato pronto non transita mai a 1, mentre il segnale di errore si porta alto durante la ricezione dell'informazione riguardante la latitudine.

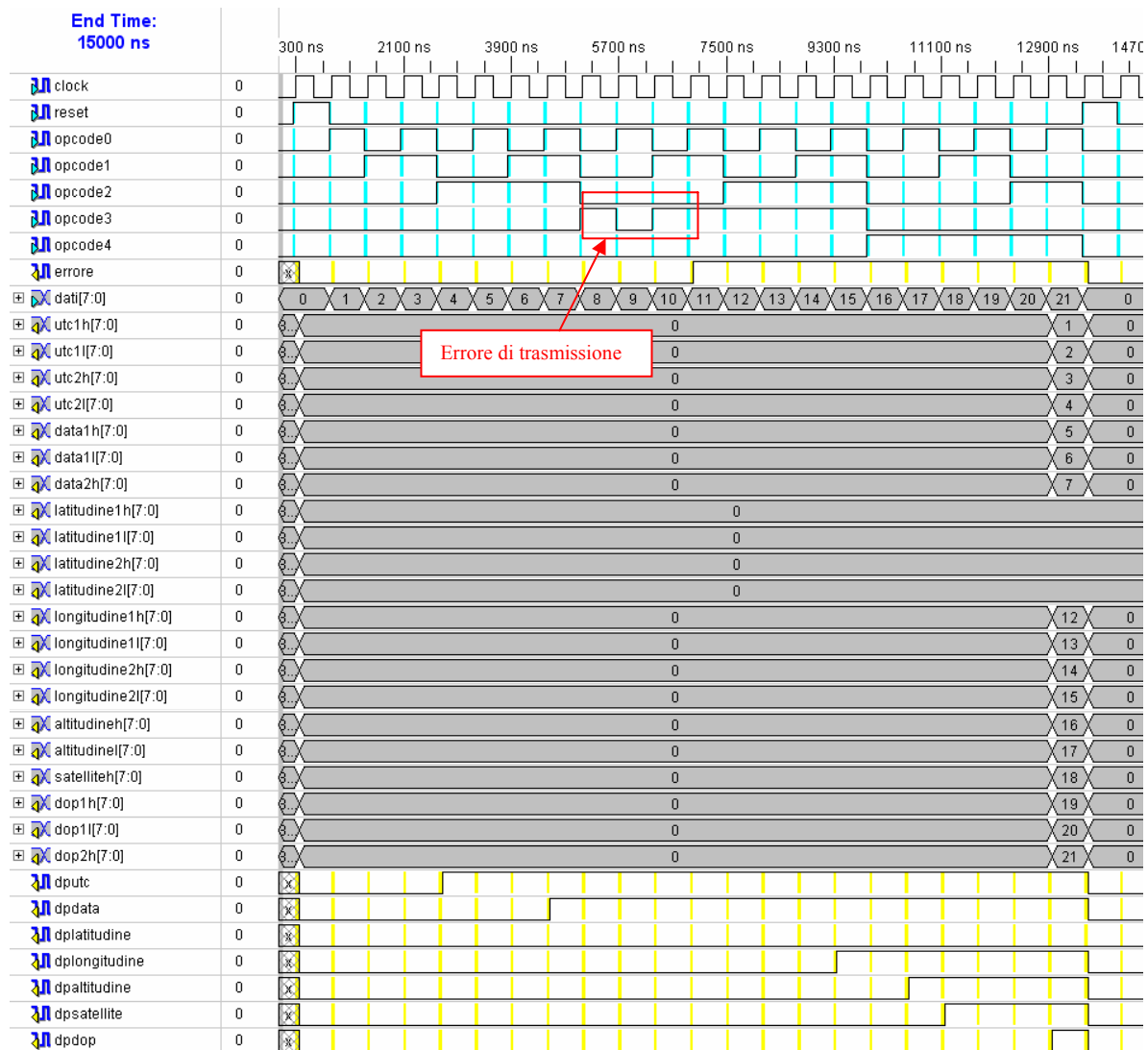
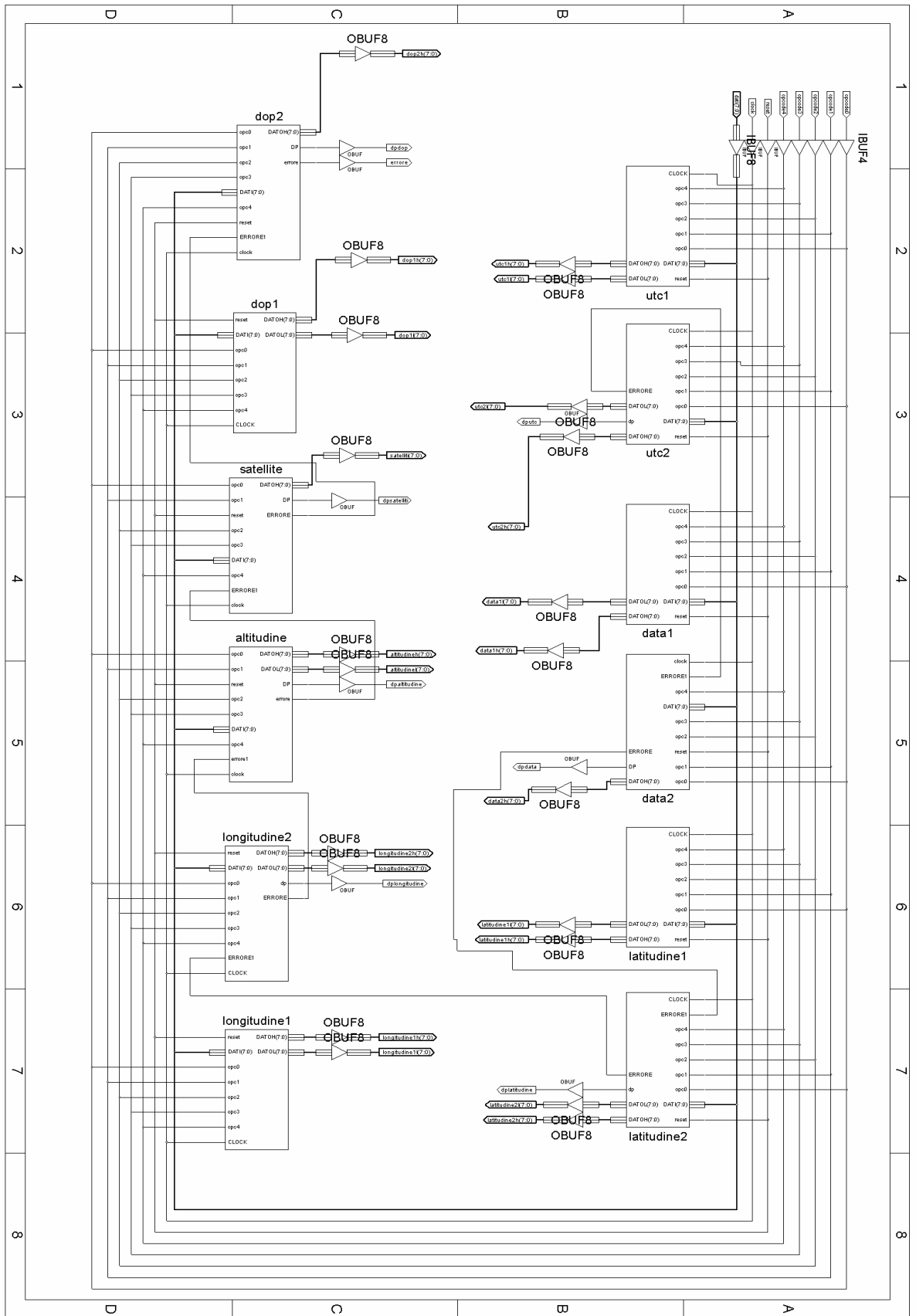


Figura 19 Simulazione del circuito con la presenza di errori

Un altro metodo di implementazione del segnale di errore, rappresentato nello **schema 8**, è ottenuto collegando serialmente le linee di errore delle PLD. In questo modo quando un errore si verifica, il segnale giunge in uscita attraversando serialmente tutte le PLD successive.

Il momento del reset, quando si presenta il segnale di errore dipende da come viene gestito nel programma della PIC. Nel programma in appendice tutti i dati vengono comunque inviati alla PLD anche se la linea d'ingresso utilizzata per l'errore si porta a 1, poiché essa viene controllata alla fine della gestione dell'interrupt.

Si può prevedere di gestire l'errore in modo tale che l'invio delle informazioni venga fermato non appena il segnale di errore è a livello logico alto. Ciò si può attuare controllando la linea d'ingresso non quando tutta l'informazione è stata inviata, ma dividendo il suo invio in tante parti quanti sono i blocchi dell'informazione. Ad esempio, prima si invia l'informazione riguardante l'UTC e si controlla l'errore: se esso è 1 si ripete l'invio di essa,



Schema 8

altrimenti si procede passando al blocco successivo, fino all'invio di tutte le informazioni.

3.10 Sviluppo del programma per le Spartan XCS10-PC84

Nello sviluppo del programma da implementare su questo tipo di PLD bisogna tener presente che non viene più utilizzato un dispositivo per ogni informazione ricevuta ma una singola PLD per tutte le informazioni, le quali non vengono più trasmesse tutte insieme verso l'esterno, ma una alla volta in base alla richiesta fatta dall'unità esterna, attuata tramite la combinazione di tre bit di richiesta.

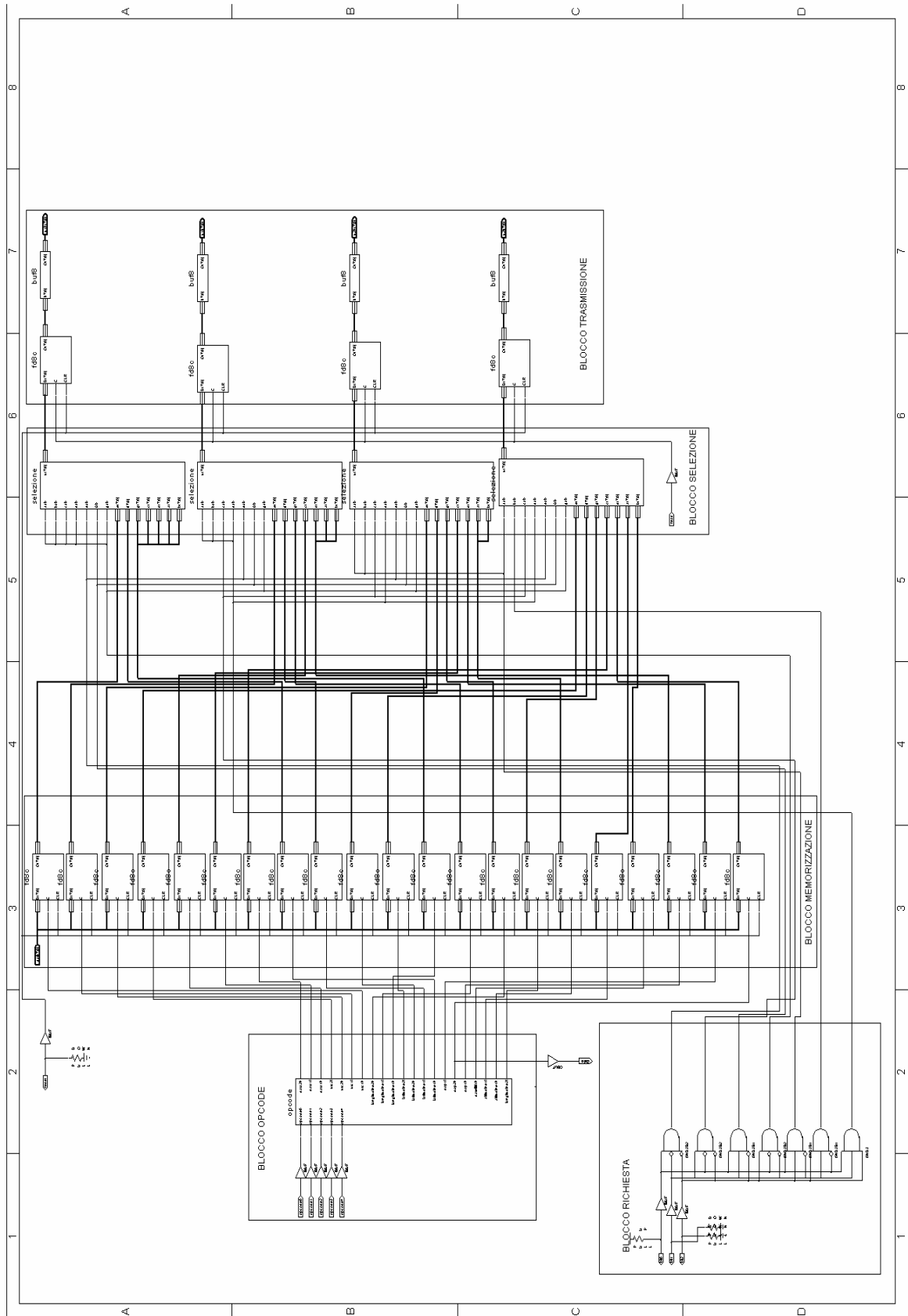
Questo nuovo tipo di programmazione permette un notevole risparmio in termini di area utilizzata, grazie al minor numero di PLD impiegate. Esso può essere attuato per la presenza del maggior numero sia di piedini I/O messi a disposizione che di gates interni disponibili, i quali consentono la memorizzazione di un programma più complesso.

Il programma realizzato, riportato nello **schema 9**, è costituito dal blocco OPCODE che serve per riconoscere l'informazione inviata dalla PIC. Il funzionamento è uguale a quello descritto per il blocco OPCODE che fa parte del programma implementato sulle XC9572, l'unica differenza è che nel blocco contenuto nel programma in analisi ci sono 21 porte AND per riconoscere tutte le informazioni inviate e non solo una parte di esse.

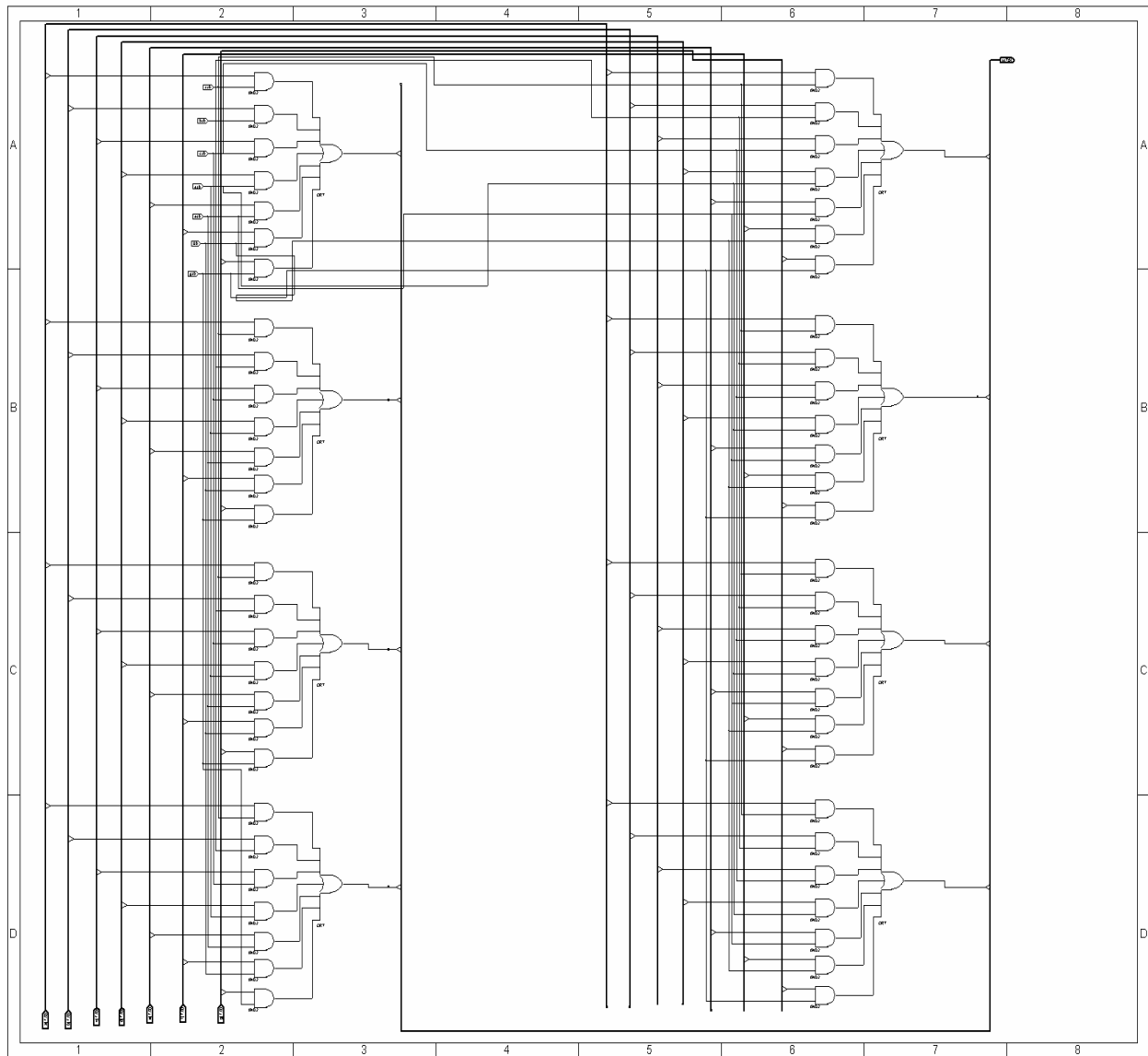
La seconda parte è costituita dal blocco MEMORIZZAZIONE, formato da una serie di 21 flip-flop, ognuno dei quali è utilizzato per mantenere in memoria un byte d'informazione. Il loro ingresso è costituito da un bus di otto bit e come clock è utilizzato uno dei segnali provenienti dal blocco OPCODE. In questo modo, sulle linee d'ingresso dei flip-flop sono presenti tutti i dati, che vengono trasferiti in uscita solo quando giunge il clock corrispondente a essi, ottenuto dall'opcode che identifica il dato per il quale il flip-flop è utilizzato.

La terza parte è rappresentata dal blocco di SELEZIONE, formato da quattro oggetti, ognuno ha il compito di portare in uscita un byte dell'informazione selezionata. Esso, come si può vedere nello **schema 10**, è costituito da sette bus (quante sono le informazioni). Ogni bit che costituisce il byte viene posto in AND con un segnale di abilitazione, che identifica l'informazione richiesta. Se il segnale è alto, in uscita alle AND avremo 1 o 0 a seconda del valore assunto dal bit che si trova sull'altro ingresso: se è 0, in uscita avremo uno 0, se è 1, il valore del bit. Tutte le uscite delle AND, corrispondenti a bit dello stesso valore, sono utilizzate come ingressi di OR, in modo che in uscita venga visualizzato il bit dell'informazione richiesta, qualunque essa sia.

L'ultima parte è composta dal blocco TRASMISSIONE, formato da una serie di quattro flip-flop, ognuno dei quali ha come ingresso l'uscita di uno degli oggetti che costituiscono il blocco SELEZIONE e come clock un segnale di



Schema 9



Schema 10

lettura proveniente dall'esterno.

Quest'ultimo blocco è utilizzato per inviare in uscita le informazioni memorizzate quando vengono richieste.

L'informazione viene selezionata dal blocco RICHIESTA, in base alla configurazione dei tre bit d'ingresso, identificata da porte AND tramite la negazione di alcuni ingressi.

Su tutti gli ingressi del programma sono previste delle resistenze di pullup o pulldown, che possono essere implementate nelle Spartan. In particolare le pulldown vengono usate per settare un ingresso a livello logico basso se fluttuante, in caso contrario le pullup. Quest'ultimo tipo di resistenza si trova solo sull'ingresso bit0, in modo che, se non viene effettuata nessuna richiesta da un unità esterna, ma c'è il segnale di lettura, in uscita, per default, è presente l'informazione riguardante l'UTC.

Nel programma è previsto, infine, un segnale di reset totale collegato a tutti i flip-flop ed un segnale di uscita con la funzione d'indicare il momento in cui tutte le informazioni sono state ricevute.

3.11 Simulazione al PC e test del programma

In **figura 20** è rappresentata la simulazione al PC del programma.

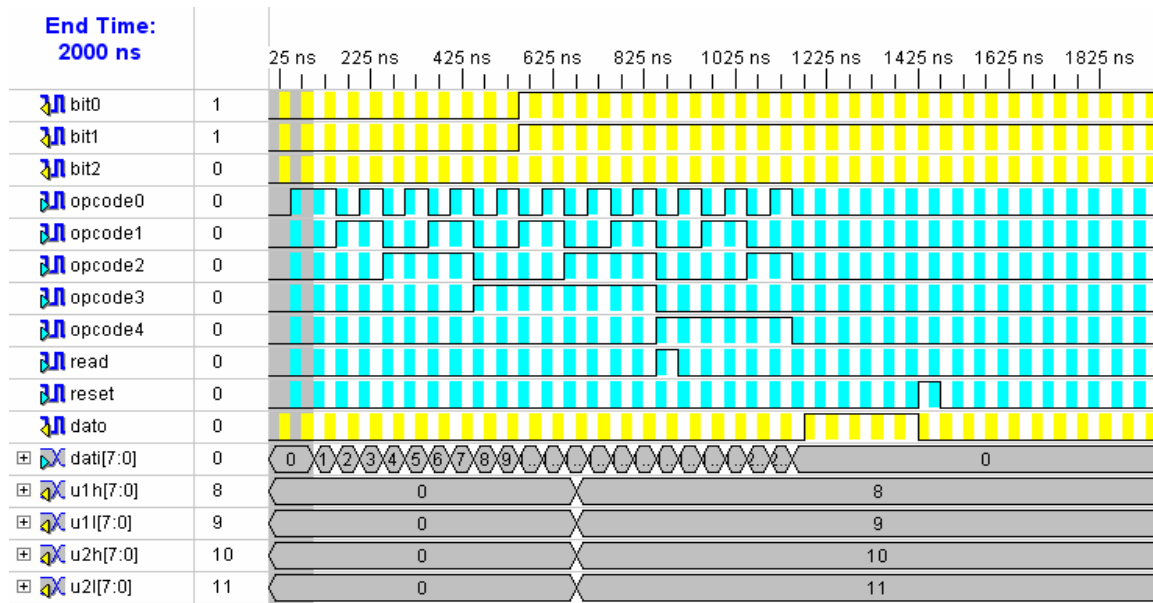
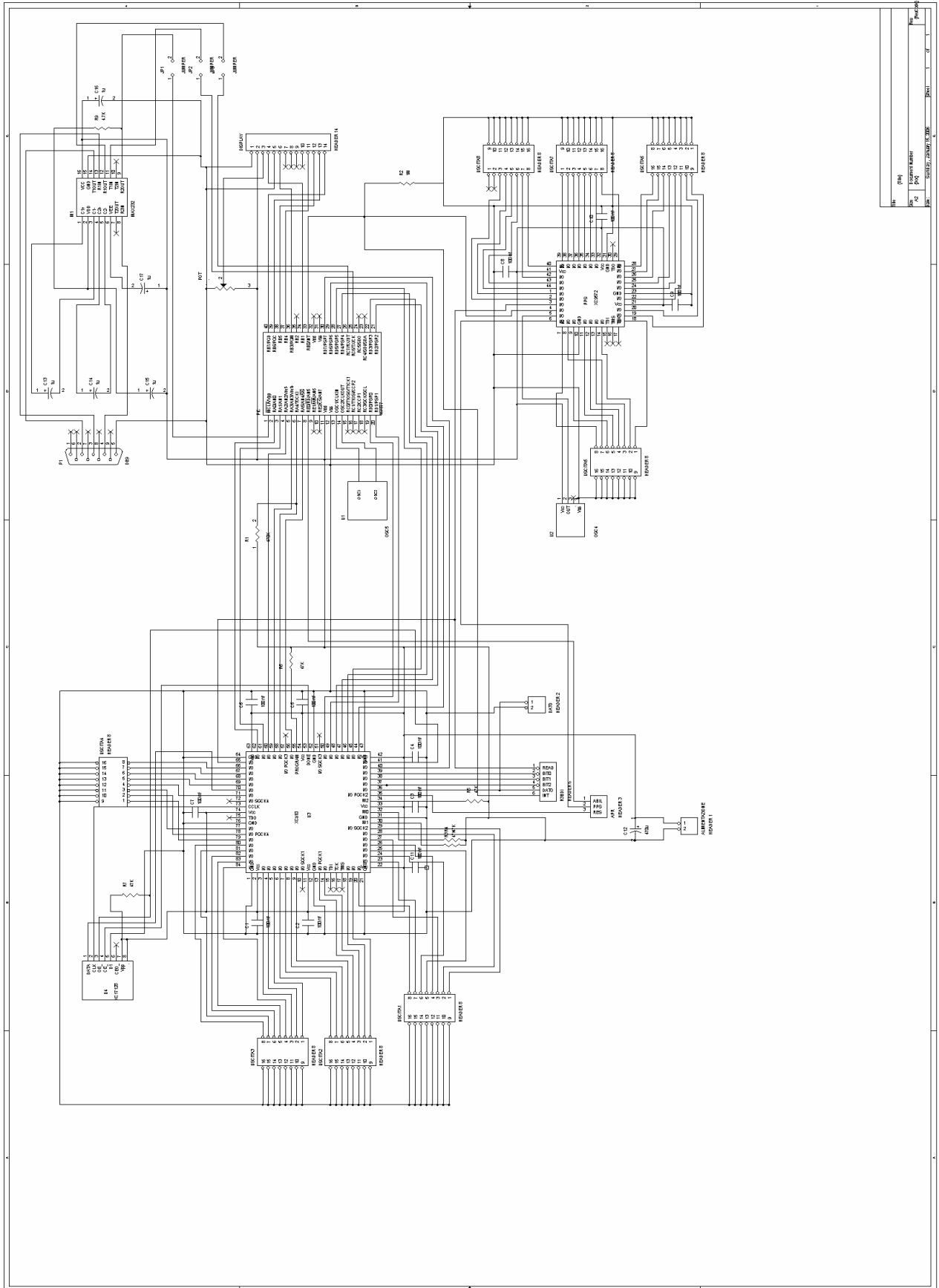


Figura 20

In questo caso, tramite i bit di selezione, viene richiesta l'informazione riguardante la latitudine che viene inviata sulle linee di uscita dopo il segnale di lettura.

Per realizzare il circuito stampato riportato in **figura 21**, necessario a verificare il corretto funzionamento del programma tramite l'analizzatore di stati logici, è stato utilizzato il programma ORCAD che consente di eseguire il layout dello **schema 11**.

Nel circuito oltre ad essere presenti tutti i dispositivi utilizzati (la PIC, la Spartan, la Xilinx 9572PC44-7 e il max 232), che consentono di realizzare il diagramma a blocchi illustrato nel capitolo 4, sono stati inseriti anche alcuni collegamenti e elementi circuitali necessari alla programmazione della PIC senza la sua estrazione dal circuito.



Schema 11

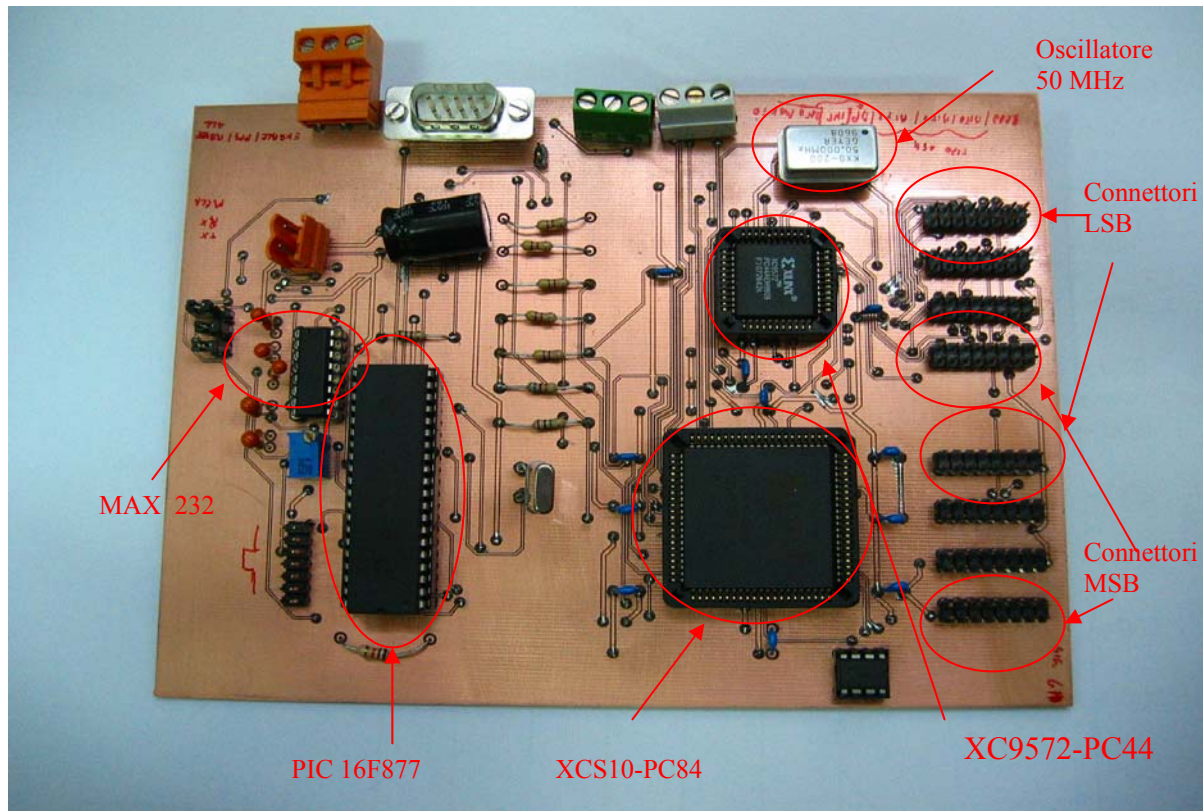


Figura 21 Circuito stampato

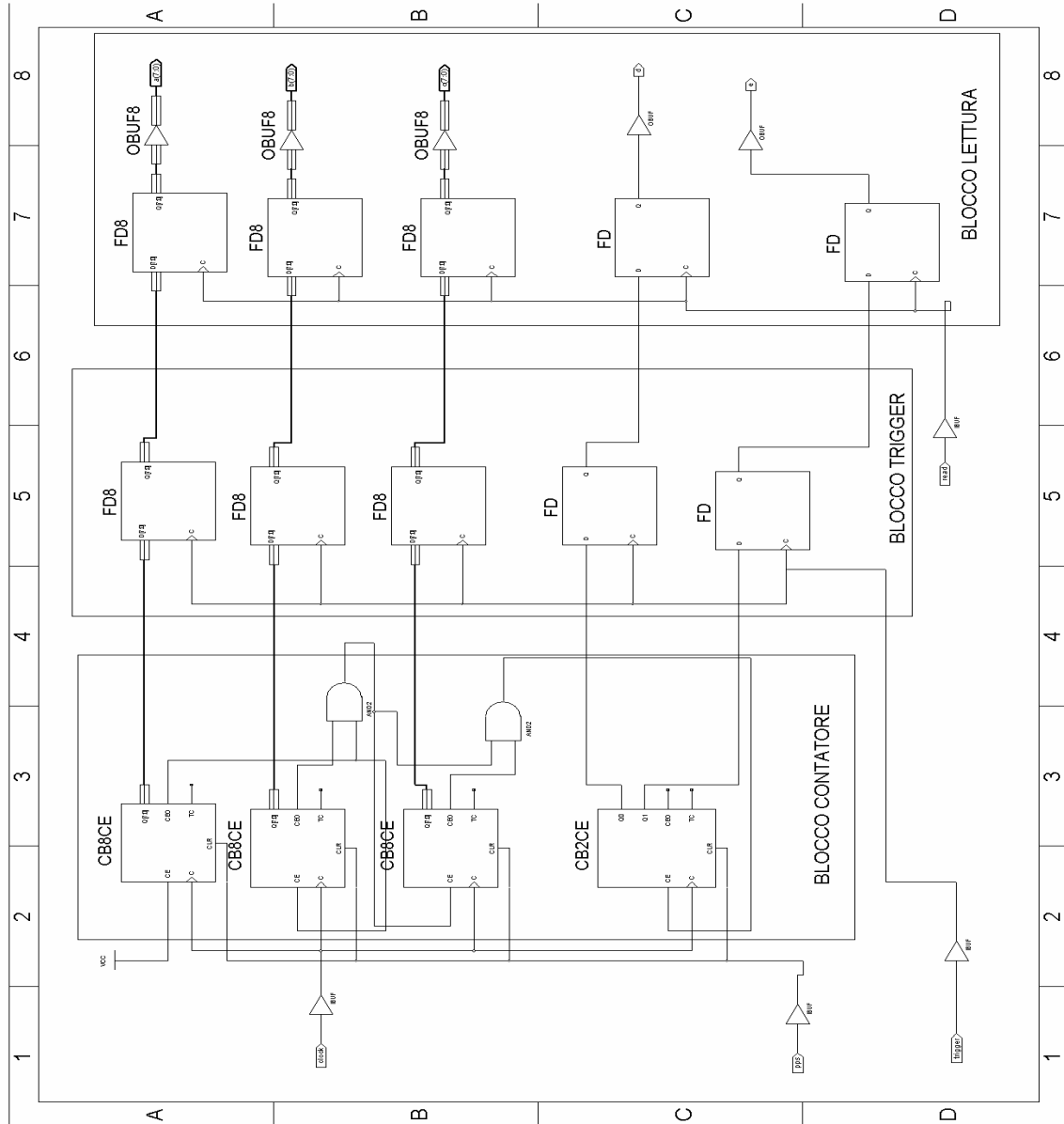
3.12 Sviluppo del programma per la gestione del tempo fino ai ns.

Nello **schema 11** e in **figura 21**, è presente una Xilinx XC9572, che riceve in ingresso il segnale PPS, il segnale di trigger, un segnale di lettura e fornisce in uscita l'informazione temporale fino ad un ordine dei ns.

Il programma implementato in essa, rappresentato nello **schema 12**, è costituito da tre blocchi: CONTATORE, TRIGGER, LETTURA.

Il blocco contatore è costituito da quattro contatori, di cui i primi tre hanno l'uscita di otto bit, l'ultimo di due. Essi sono collegati a cascata, in modo tale che nel momento in cui il contatore che si trova più in alto, ha raggiunto il massimo numero di conteggi, quello successivo ne effettui il primo; dopo di che il contatore iniziale si azzerà ed il processo viene ripetuto. La stessa operazione viene eseguita fra il secondo contatore e il terzo e fra il terzo e l'ultimo. In questo modo possono essere eseguiti fino a 2^{26} conteggi, ma in realtà il numero massimo eseguito è di 50×10^6 , in quanto giunti a questa cifra i contatori vengono resettati.

L'avanzamento del conteggio avviene tramite un segnale collegato al clock di cui questi sono dotati, questo segnale è fornito da un quarzo di 50 MHz, mentre il segnale di reset è costituito dal PPS.



Schema 12

In questo modo, si ottiene il frazionamento del tempo che intercorre tra un

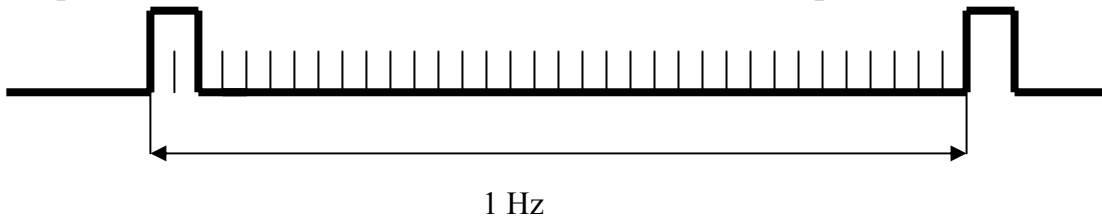


Figura 22 Frazionamento del PPS

fronte di salita dell'impulso PPS e l'altro (**figura 22**), in 50×10^6 intervalli, ognuno dei quali ha durata di 2 ns.

Il secondo blocco di TRIGGER è utilizzato per memorizzare il numero di conteggi eseguiti fino a quando non interviene un segnale generato esternamente (lo stesso di quello che costituisce l'interrupt per la PIC).

Da una serie di flip – flop che hanno come clock il segnale di trigger e come ingressi i dati provenienti dal blocco CONTATORE, in modo che all'arrivo del clock i dati in ingresso vengano memorizzati in uscita.

L'ultimo blocco, quello di LETTURA, è anche esso costituito da una serie di flip – flop con il compito di inviare in uscita i dati immagazzinati in memoria quando interviene il segnale di lettura.

Nella gestione del segnale di reset dei contatori bisogna tener presente che essi mantengono la loro uscita fissa a zero, indipendentemente dal segnale di clock applicato quando l'ingresso riservato al reset è mantenuto ad un livello logico alto. Ciò costituisce un problema in quanto il conteggio dovrebbe iniziare il più vicino possibile alla transizione del PPS da 0 a 1, ma poiché il livello logico alto del segnale ha una durata dell'ordine dei μs , il contatore viene abilitato a contare con notevole ritardo.

Per risolvere questo problema, il segnale PPS non deve essere indirizzato direttamente verso la PLD ma deve essere utilizzato per generare un impulso sincrono con il suo fronte di salita.

Ciò è realizzato schematicamente come mostrato in **figura 23**,

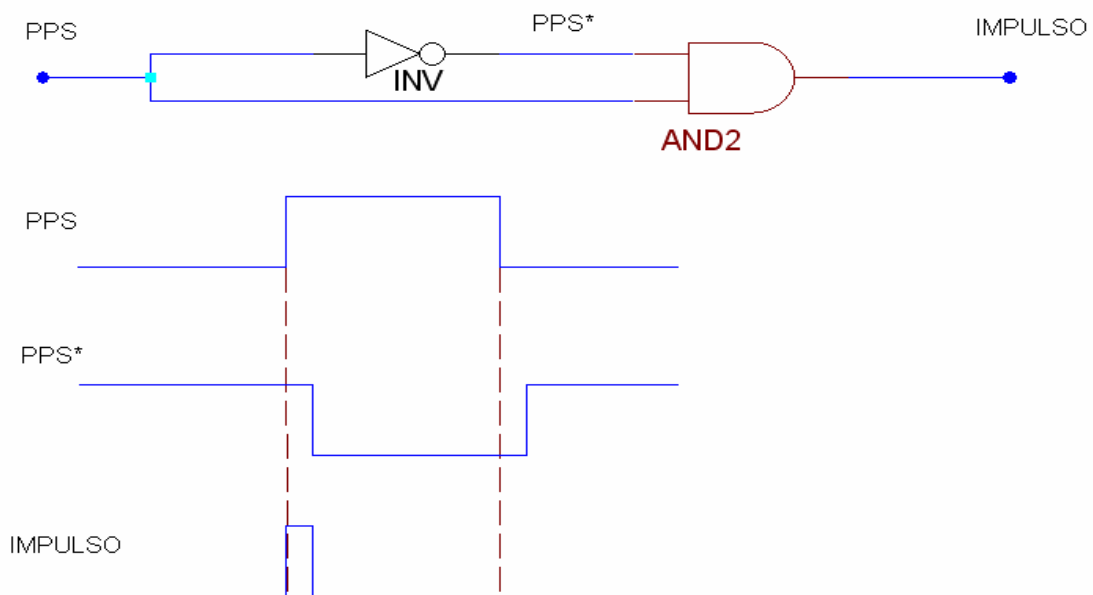


Figura 23 Generatore di impulsi

L'invertitore produce un ritardo di pochi ns, cosicché le transizioni di PPS* avvengono poco dopo quelle di PPS. La porta AND produce uno stretto impulso di uscita, che è alto solo pochi ns, quando PPS e PPS* sono entrambi alti. Il risultato è uno stretto impulso durante il fronte d'onda positivo del PPS.

L'integrato utilizzato per far questo è il 7400, su cui sono implementate una serie di porte NAND, i cui ingressi e uscite sono state collegati per ottenere un circuito che dà un risultato equivalente a quello in **figura 23**.

L'impulso così ottenuto è utilizzato come segnale di reset dei contatori.

3.13 Fonti di errore

La fase della progettazione del circuito per fornire il tempo dell'ordine dei ns è la più critica da un punto di vista della gestione dell'errore. Esso, infatti, può essere generato da diverse fonti:

- 1) Imprecisa sincronizzazione fra il fronte di salita del PPS proveniente dal GPS e il fronte di salita del clock del ricevitore che deve dare in uscita il segnale per inviarlo alla PLD (errore di quantizzazione).
- 2) Variazione della frequenza di oscillazione del quarzo utilizzato.
- 3) Arrivo del segnale di trigger in un istante non corrispondente ad un fronte di salita del clock.
- 4) Imprecisa sincronizzazione fra il fronte di salita del clock generato dal quarzo collegato al PLD e quello del PPS.

Essi sono rappresentati tutti in **figura 24**.

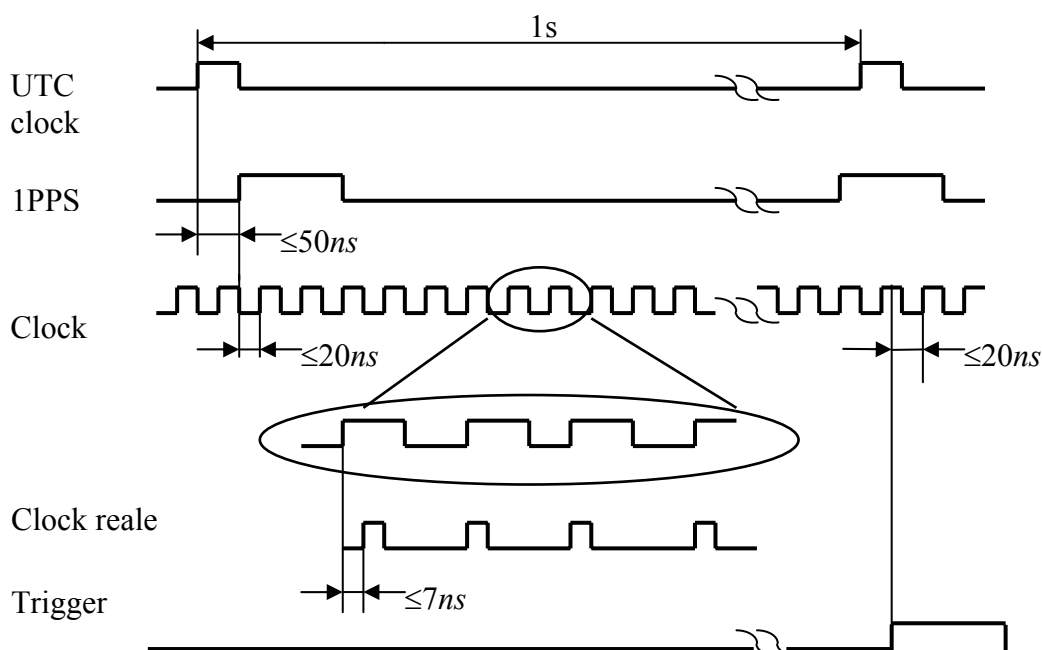


Figura 24 Errori del sistema

Gli errori dovuti agli ultimi due punti non possono essere eliminati. Vediamo come è invece possibile ridurre quelli derivanti dalle prime due fonti.

L'errore dovuto al punto 1) è stato ridotto utilizzando il ricevitore GPS Resolution T prodotto dalla Trimble, il quale garantisce una distribuzione dell'impulso PPS all'interno di una σ di $15ns$.

Per quanto riguarda il punto 2), bisogna considerare che la precisione di cui abbiamo bisogno è di $\pm 10ns$, ciò significa che, utilizzando un oscillatore da $50MHz$ abbiamo bisogno di un'accuratezza Δf di $\pm 0.5Hz$, quello che è stato adottato ne ha una di $\pm 0.35Hz$.

Si sarebbe potuto anche utilizzare un quarzo con una frequenza di oscillazione maggiore, il che comporta una minore ampiezza di ogni singolo intervallino temporale in cui viene diviso il PPS e una maggiore precisione nel tempo fornito. In commercio però, i quarzi a maggiore frequenza che garantiscono un'accuratezza così elevata hanno un costo eccessivo, e comunque, cercare di ridurre quest'errore è inutile, poiché nella propagazione degli errori che intervengono in questo dispositivo il termine dominante è quello dato dal PPS del GPS. Infatti posto:

- $E_{GPS} \cong \pm 15ns$ l'errore associato al PPS del Resolution T.
- $E_{OSC} \cong \pm 7ns$ l'errore dovuto alla precisione dell'oscillatore.
- $E_{TRIG} \cong \pm 20ns$ l'errore dovuto all'imprecisa sincronizzazione fra il trigger e il segnale di clock.
- $E_C \cong \pm 20ns$ l'errore dovuto all'imprecisa sincronizzazione fra clock e PPS.

Assumendo che gli errori eguano una distribuzione normale, risulterà un errore totale pari a:

$$E_{tot} = \sqrt{E_{GPS}^2 + E_{OSC}^2 + E_{TRIG}^2 + E_C^2} = \pm 32,77ns$$

Supponendo di annullare il fattore E_{OSC} , E_{tot} diminuisce fino ad un valore di $\pm 32,01ns$, abbiamo dunque, un miglioramento di $0,76ns$, valore troppo basso per giustificare il notevole aumento di costo da sopportare.

3.14 Simulazione al PC e test del programma

La simulazione di questo programma, riportata in **figura 25**, è stata fatta non fornendo i valori esatti di frequenza del segnale di CLOCK e del PPS, poiché troppo elevato il primo e molto basso il secondo.

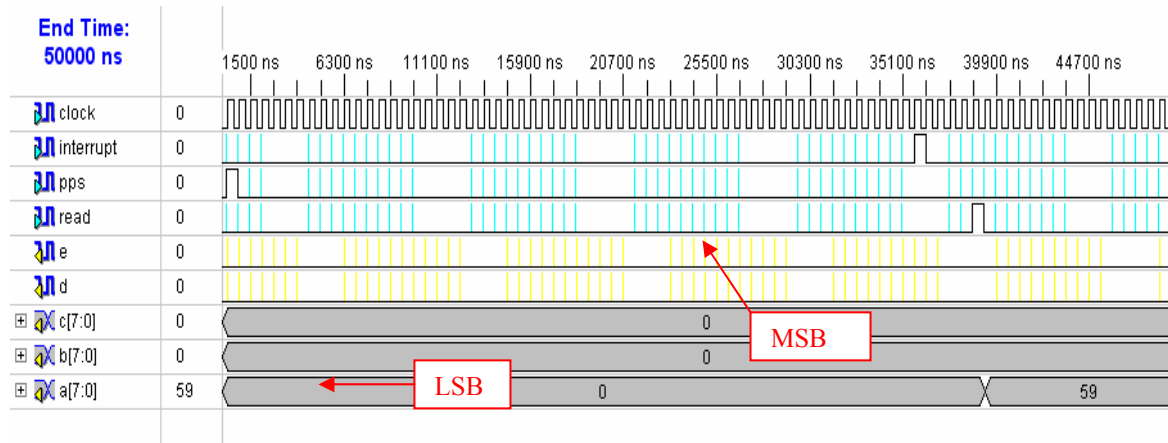


Figura 25 Simulazione del programma di conteggio

Il risultato comunque è quello previsto: il numero di transizioni che il CLOCK effettua, nell'intervallo di tempo che intercorre dall'impulso generato dal segnale di trigger (chiamato interrupt nella simulazione), sono 59, visualizzate in uscita dopo il segnale di lettura.

Per testare il programma con il circuito stampato realizzato è stato utilizzato il generatore di funzioni, il generatore di dati e l'analizzatore di stati logici.

Il generatore di funzioni è collegato al generatore di dati, per simulare il trigger di cui quest'ultimo necessita per poter inviare i dati in uscita, cioè: un impulso ripetuto ogni secondo che simula il PPS, due altri segnali che simulano il segnale di trigger e di lettura. Il risultato è successivamente visualizzato sull'analizzatore di stati logici.

CONCLUSIONI

Al fine di ottenere la sincronizzazione temporale fra sistemi di controllo remoti, si è fatto uso del GPS.

Nell'usare tale apparato sono state affrontate quattro problematiche:

- Estrazione delle informazioni inviate da un ricevitore per GPS nello standard NMEA-0183.
- Invio delle informazioni estratte verso dispositivi esterni, quando ne viene fatta richiesta, per la loro parallelizzazione, rendendole adatte ad essere interfacciate con un bus standard (VME).
- Parallelizzazione delle informazioni.
- Messa a disposizione dell'informazione temporale con una precisione dell'ordine dei ns.

Le prime due sono state affrontate tramite l'uso di una PIC, creando un programma adatto a rintracciare le informazioni a noi utili provenienti dal ricevitore GPS e quindi, a gestire dati d'ingresso seriali.

Nella fase di trasmissione delle informazioni, è stata sfruttata una particolare caratteristica della PIC: la gestione dell'interrupt. Questa permette di inviare direttamente informazioni quando un segnale esterno interviene evitando così di sprecare tempo nell'interrogare lo stato della linea su cui esso è presente. La terza e la quarta problematica sono state affrontate, tramite l'uso delle PLD.

Tutti i programmi realizzati sono stati simulati al PC per garantirne il corretto funzionamento e, successivamente, si è provveduto a effettuare una verifica pratica, costruendo diversi prototipi e visualizzando le informazioni ottenute attraverso un analizzatore di stati logici.

APPENDICE A

MISURE EFFETTUATE

Una scelta fatta nel programma della PIC è di non trasmettere nessun dato quando avviene un interrupt ed il ricevitore non riesce ad agganciare un numero di satelliti superiore a tre.

Questa scelta si basa su delle misure, che hanno come oggetto la variazione della precisione con cui il ricevitore GPS fornisce il segnale PPS quando non riesce ad agganciare nessun satellite.

Esse sono eseguite visualizzando sull'oscilloscopio il PPS fornito da due ricevitori, il Resolution T che è quello adottato, ed il Lassen iQ, entrambi della Trimble, ed impostando il trigger dell'oscilloscopio sulla variazione del PPS fornito dal secondo dei due ricevitori che quindi fa da riferimento (**figura 26**).

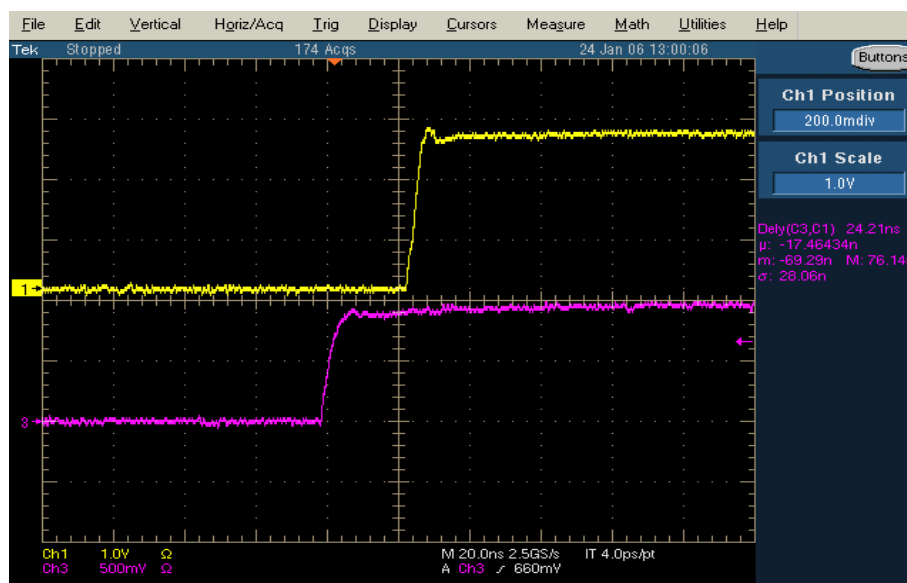


Figura 26

Sono effettuate due diverse misure, la prima quando entrambi i ricevitori agganciano dei satelliti, la seconda quando il Resolution T non riesce a vederne nessuno. Per entrambe sono state effettuate 2000 acquisizioni. In **figura 27** è rappresentato il grafico contenente i valori ottenuti nel primo caso. Come si può vedere, la distribuzione che essi assumono è di tipo normale con una deviazione standard di $39,519ns$, compatibile con quella che deriva dalla somma in quadratura delle deviazioni dichiarate per il Resolution T ($15ns$) e il Lassen iQ ($52ns$).

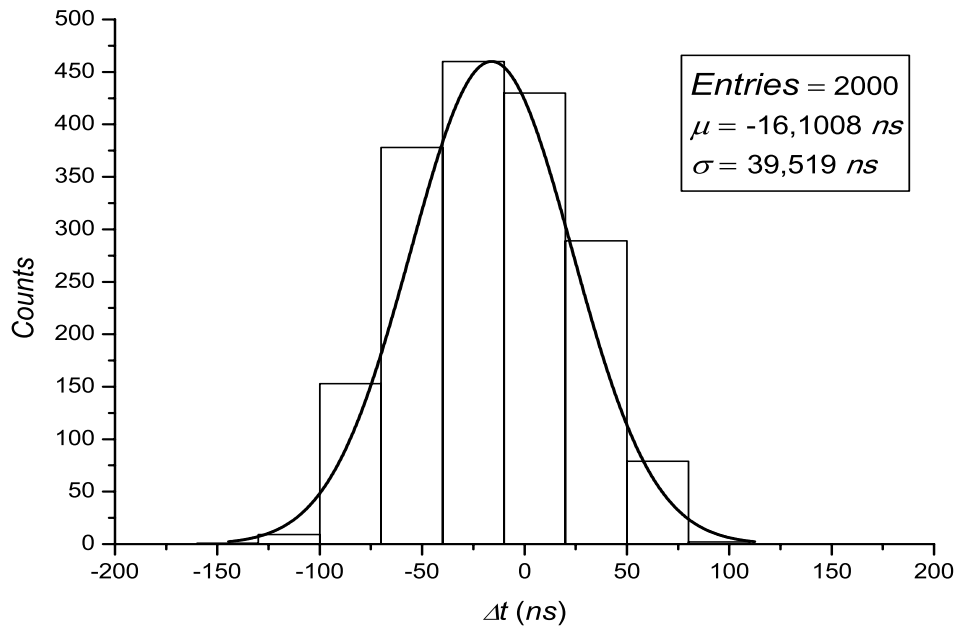


Figura 27 Resolution T riesce ad agganciare i satelliti

In **figura 28**, è rappresentato il risultato ottenuto dalle misure effettuate quando il Resolution T non vede satelliti.

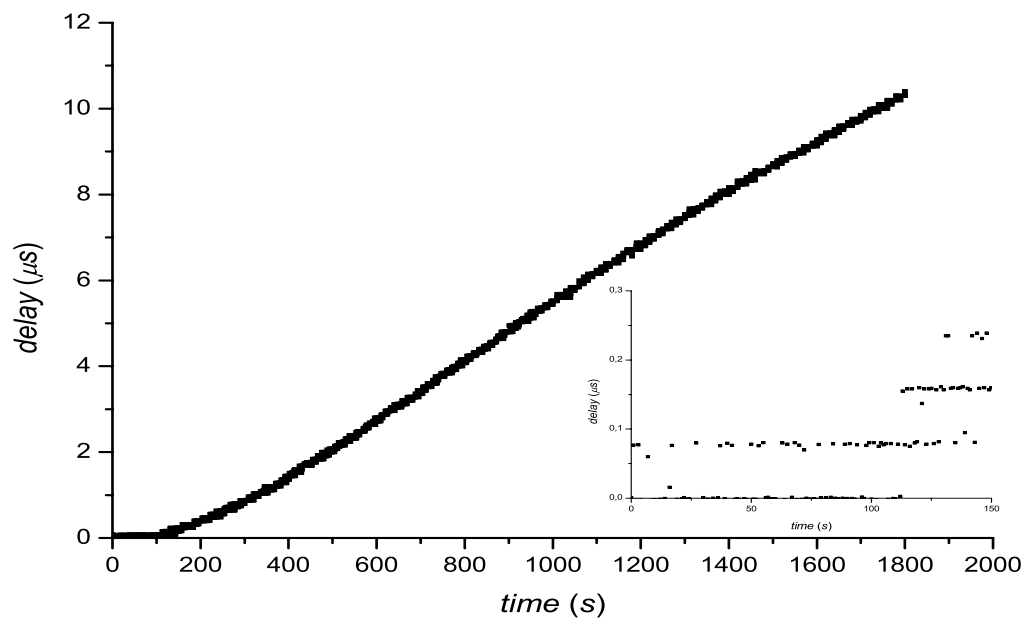


Figura 28 Il Resolution T non riesce ad agganciare i satelliti

Come si può vedere nel riquadro in basso, fino ai 100s, i valori di ritardo si mantengono inferiori a $0,1\mu\text{s}$, questo fa sì che in questo range di tempo, il Resolution T garantisce il segnale PPS in modo affidabile, dopo di che il ritardo tende ad aumentare di circa 600ns ogni 100s. In questo caso infatti il ricevitore non fornisce in uscita il segnale PPS derivato dai satelliti, ma il suo clock interno.

APPENDICE B

.Programma della PIC

Il programma realizzato prevede due diverse versioni, qui di seguito, sono riportate in nero le istruzioni comuni a entrambe, in verde quelle relative solo alla prima versione, in celeste quelle riguardanti la seconda.

```

*****
* Name      : GPS.BAS                                     *
* Author    : GIUSEPPE COCCIOLO                         *
* Date      : 15/12/2005                                 *
*****

```

```

DEFINE  OSC 20                                     ‘Definizione dell’oscillatore
INCLUDE  “modedefs.bas”
ADCON1 = 7                                         ‘Impostazioni delle uscite come digitali

```

```

ON INTERRUPT GOTO TRASMISSIONE                    ‘Dichiarazione dell’interrupt
INTCON = %10010000                                ‘Abilitazione dell’interrupt proveniente
                                                    ‘dalla linea RB0

```

```

TRISC = %10000000                                ‘Impostazione delle porte utilizzate
TRISA=0                                           ‘come ingressi o uscite
TRISD = 0
TRISB = %00000001
TRISE =%001

```

```

DEFINE  LCD_DREG PORTB                            ‘Definizione delle porte utilizzate per
DEFINE  LCD_DBIT 4                                ‘la gestione del display LCD
DEFINE  LCD_BITS 4
DEFINE  LCD_RSREG PORTB
DEFINE  LCD_RSBIT 1
DEFINE  LCD_EREG PORTB
DEFINE  LCD_EBIT 3
DEFINE  LCD_LINES 2

```

```

ORE          VAR BYTE                            ‘Definizione delle variabili
MINUTI       VAR BYTE
SECONDI     VAR BYTE
SECINDI2    VAR BYTE
TEMPOL     VAR WORD
TEMPOH     VAR WORD
TEMPO      VAR WORD
LA1        VAR BYTE
LA2        VAR BYTE
LA3        VAR WORD

```

```

LATH      VAR WORD
LATL      VAR WORD
NS        VAR BYTE[1]
LO1       VAR WORD
LO2       VAR WORD
LO3       VAR WORD
LONH      VAR WORD
LONL      VAR WORD
EW        VAR BYTE[1]
ALTITUDINE VAR WORD
SATELLITI VAR BYTE
P         VAR BYTE
P1        VAR BYTE
H         VAR BYTE
V         VAR BYTE
V1        VAR BYTE
GIORNO    VAR BYTE
GIORNO2   VAR BYTE
MESE      VAR BYTE
ANNO      VAR BYTE
DATI      VAR BYTE[22]
I         VAR BYTE
A         VAR BIT
CONTROLLO VAR BIT
W         VAR PORTE.0
U         VAR PORTE.1
CLOCK    VAR PORTA.5
OPCOD    VAR PORTC
IN       VAR PORTC.7
OUT      VAR PORTC.6
USCITA   VAR PORTD
    
```

```

LCDOUT $FE,1 "DATI RILEVATI"
LCDOUT $FE, $C0, "DAL GPS"
    
```

‘Controllo sul corretto funzionamento
‘del display

```
SEROUT2 OUT, 188, [16, 142, 38, 16, 3]
```

‘Memorizza nel ricevitore le ultime
‘impostazioni settate

INIZIO:

```
IF W=1 THEN
```

```
U=1
```

```
U=0
```

```
GOTO INT
```

```
ENDIF
```

```
U=1
```

```
U=0
```

```
CONTROLLO = 0
```

‘Estrazione delle informazioni provenienti dal GPS

```
SERIN2 IN, 188, [WAIT("GSA,"), SKIP 28, DEC P, DEC P1, DEC H]
```

```
SERIN2 IN, 188, [DEC H1, DEC V, DEC V1]
```

```
SERIN2 IN, 188, [WAIT("RMC,"), SKIP 40, WAIT(",")]
```

```

SERIN2 IN, 188, [DEC2 GIORNO, DEC2 MESE, DEC2 ANNO]
SERIN2 IN, 188, [WAIT("GGA,"), DEC2 ORE, DEC2 MINUTI]
SEIRN2 IN, 188, [DEC2 SECONDI, WAIT(", "), DEC2 LA1, DEC2 LA2]
SERIN2 IN, 188, [DEC4 LA3, WAIT(", "), NS, WAIT(", "), DEC3 LO1]
SERIN2 IN, 188, [DEC2 LO2, DEC4 LO3, WAIT(", "), EW]
SERIN2 IN, 188, [WAIT(", "), WAIT(", "), DEC2 SATELLITI]
SERIN2 IN, 188, [WAIT(", "), WAIT(", "), DEC ALTITUDINE]

```

```

SECONDI2 = SECONDI
GIORNO2 = GIORNO
CONTROLLO = 1

```

‘Variabili per il controllo
‘dell’esatto valore dell’UTC
‘e della data

‘Controllo sull’affidabilità delle informazioni

```

IF SATELLITI > 3 AND SATELLITI < 13 THEN

```

‘Visualizzazione delle informazioni

```

LCDOUT $FE, 1, "UTC: ", DEC2 ORE, DEC2 MINUTI, DEC2 SECONDI
PAUSE 1000

```

```

LCDOUT $FE, 1, DEC2 LA1, DEC2 LA2, DEC4 LA3, " ", NS
PAUSE 1000

```

```

LCDOUT $FE, 1, DEC2 LO1, DEC2 LO2, DEC4 LO3, " ", EW
PAUSE 1000

```

```

LCDOUT $FE, 1, DEC2 SATELLITI, " ", DEC ALTITUDINE
PAUSE 1000

```

```

LCDOUT $FE, 1, DEC P, ".", DEC P1, DEC H, ".", DEC H1, " ", DEC V, ".",
DEC V1
PAUSE 1000

```

```

LCDOUT $FE, 1, DEC2 GIORNO, "-", DEC2 MESE, "--", DEC2 ANNO
PAUSE 1000

```

```

ENDIF

```

```

IF SATELLITI > 13 THEN

```

```

LCDOUT $FE, 1, "SATELLITI ASSENTI "

```

```

LCDOUT $FE, $C0, "RILEVATI : 0"

```

```

ENDIF

```

```

IF SATELLITI < 3 THEN

```

```

LCDOUT $FE, 1, "SATELLITI ASSENTI "

```

```

LCDOUT $FE, $C0, "RILEVATI : ", DEC2 SATELLITI

```

```

ENDIF

```

```

GOTO INIZIO

```

```

END

```

```

INT:

```

```

DISABLE

```

```

TRASMISSIONE :

```

```

IF SATELLITI > 3 AND SATELLITI < 13 THEN

```

‘Inizio programma per la
‘gestione dell’interrupt

```

P = ( P * 100 ) + P1

```

```

H = ( H * 100 ) + H1

```

```

V = ( V * 100 ) + V1

```



```

TEMPOL = 0
TEMPOH = 0
‘Controllo sull’esatto valore dell’UTC
IF ( CONTROLLO = 0 AND SECONDI = SECONDI2 ) THEN
A = 1
ELSE
A = 0
ENDIF
TEMPOH = 3600 ** ORE
TEMPOL = 3600 * ORE
TEMPO = 60 * MINUTI + SECONDI + A
IF 65535 – TEMPOL <= TEMPO THEN
TEMPOH = TEMPOL + TEMPO
‘Immagazzinamento dell’UTC
‘in due variabili Word

LATH = ( LA1 * 60 + LA2 ) ** 10000
LATL = ( ( LA1 * 60 ) + LA2 ) * 10000
IF 65535 – LATL <= LA3 THEN
LATH = LATH + 1
ENDIF
LATL = LATL + LA3
IF NS = “ N “ THEN
LATH =LATH + 32768
ENDIF
‘Immagazzinamento della latitudine
‘in due variabili Word

LONH = ( LO1 * 60 + LO2 ) ** 10000
LONL = ( ( LO1 * 60 ) + LO2 ) * 10000
IF 65535 – LONL <= LO3 THEN
LONH = LONH + 1
ENDIF
LONL = LONL + LO3
IF EW = “ E “ THEN
LONH =LONH + 32768
ENDIF
‘Immagazzinamento della longitudine
‘in due variabili Word

‘Controllo sull’esatto valore della data
IF ( CONTROLLO = 0 AND TEMPOL = 0 AND TEMPOH = 0 AND GIORNO =
GIORNO2 ) THEN
GIORNO = GIORNO2 + 1
ENDIF
‘Immagazzinamento delle informazioni in un vettore a gruppi di otto bit
DATI[1] = TEMPOH.Byte1
DATI[2] = TEMPOH.Byte0
DATI[3] = TEMPOL.Byte1
DATI[4] = TEMPOL.Byte0

DATI[5] = GIORNO
DATI[6] = MESE
DATI[7] = ANNO

DATI[8] = LATH.Byte1
DATI[9] = LATH.Byte0

```

DATI[10] = LATL.Byte1
 DATI[11] = LATL.Byte0

DATI[12] = LONH.Byte1
 DATI[13] = LONH.Byte0
 DATI[14] = LONL.Byte1
 DATI[15] = LONL.Byte0

DATI[16] = ALTITUDINE.Byte1
 DATI[17] = ALTITUDINE.Byte0

DATI[18] = SATELLITI

DATI[19] = P
 DATI[20] = H
 DATI[21] = V

WHILE W=1
WEND

U=1
 U=0

FOR I = 1 **TO** 21
 USCITA = DATI[I]
 OPCOD = I
 CLOCK = 1
 CLOCK = 0
NEXT I
ELSE
 U=1
 U=0
ENDIF
 INTCON.1 = 0
RESUME
ENABLE

‘Invio delle informazioni

APPENDICE C

Programma in LabView

Inizialmente, prima dello sviluppo dei programmi descritti, è stata realizzata un'interfaccia in LabView con lo scopo di apprendere il linguaggio NMEA descritto nel capitolo 4.

In questa appendice viene descritto brevemente il funzionamento di alcuni blocchi del programma.

Tramite questa interfaccia (**figura 29**), è possibile settare le caratteristiche di velocità di trasmissione della porta seriale con cui si comunica con il ricevitore GPS, il data bit, il parità bit, lo stop bit. I dati identificati che vengono visualizzati sono: UTC, numero dei satelliti, latitudine, longitudine, altezza e data.

L'UTC viene visualizzato istante per istante, oltre che in formato stringa, su una matrice a LED, al fine di riprodurre il dato in formato binario, con il quale è inviato verso le unità esterne nel programma sviluppato per la PIC.

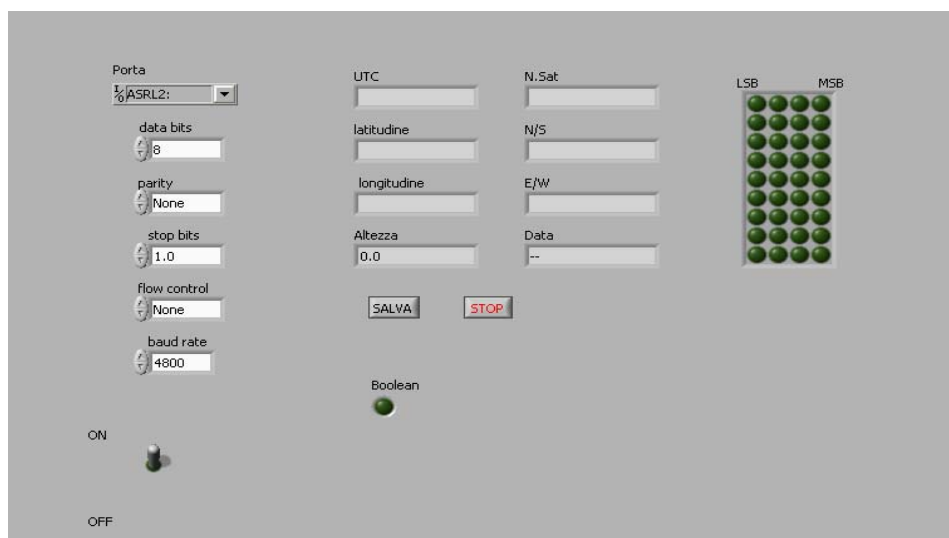


Figura 29 Interfaccia sviluppata in LabView

Sull'interfaccia è previsto un tasto salva, che consente di memorizzare dei dati di maggiore interesse in un file di tipo .txt.

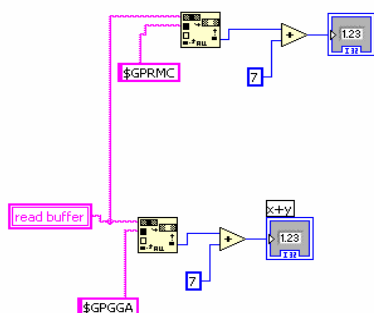


Figura 30

Alcuni dei blocchi con cui è stato realizzato il programma sono rappresentati nelle successive figure. Lo schema in **figura 30** consente di rintracciare nel messaggio ricevuto, le stringhe GPRMC e GPGLA.

Una struttura uguale viene utilizzata per

localizzare anche le altre stringhe di interesse.

Nella **figura 31**, è mostrato il blocco che, una volta identificata la stringa in cui l'UTC è contenuto, permette di estrarlo e visualizzarlo sulla matrice a LED, tramite il subVI 1, il quale, presa la stringa intercettata, la fraziona in vettori, al fine di poterla rappresentare sulla matrice.

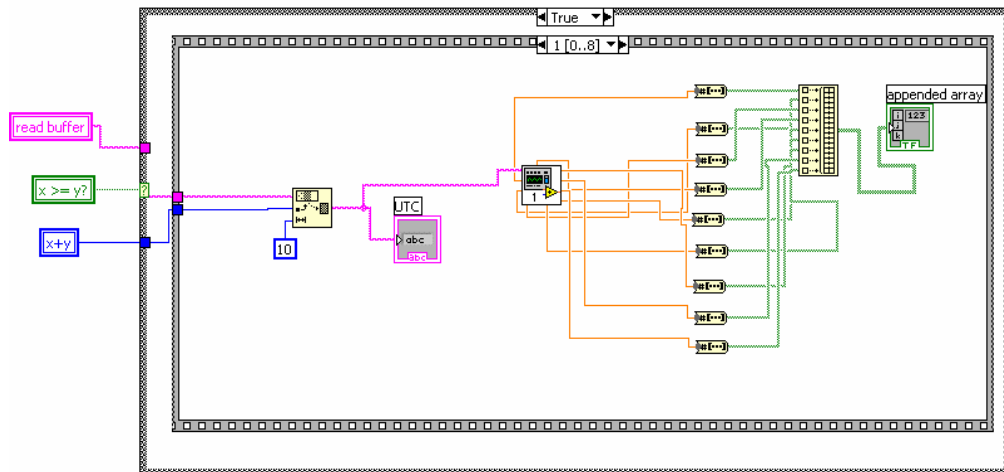


Figura 31

L'implementazione del tasto salva è ottenuta tramite lo schema a blocchi di **figura 32**.

Tutti i dati ricevuti vengono convogliati in una tabella ed una volta premuto il tasto salva, vengono memorizzati nel file gps.txt.

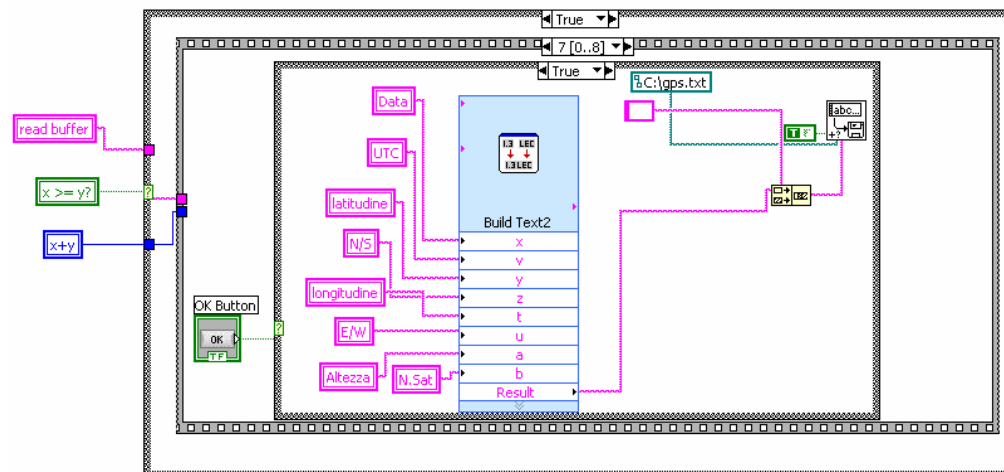


Figura 32

Nello schema di **figura 32** è presente il subVI 2, il quale riceve in ingresso la stringa inviata dal ricevitore GPS, insieme al numero che indica la posizione dove l'informazione a cui siamo interessati è localizzata (nel nostro caso la

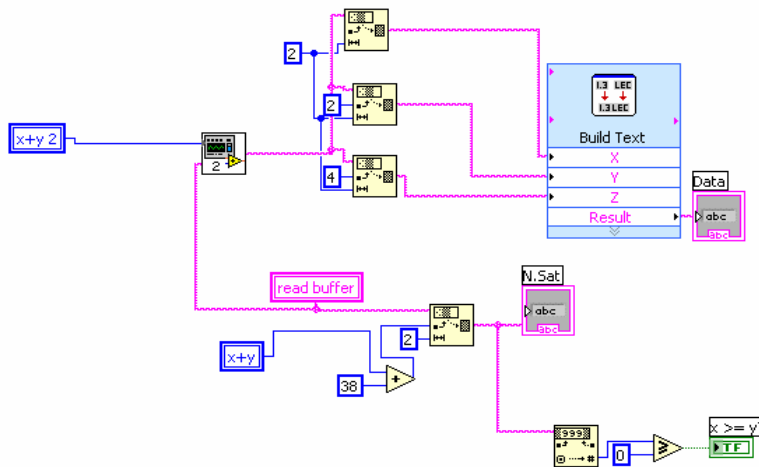


Figura 32

data) e fornisce in uscita la parte di stringa contenente l'informazione che interessa. Infine questa viene scomposta nelle tre parti di cui la data è formata: giorno, mese e anno; successivamente tramite il blocco build text viene visualizzata nel formato gg-mm-aa.

BIBLIOGRAFIA

- [1] Antonino Zichichi “Progetto: “La scienza nelle scuole” EEE Extreme Energy Events” – Università di Bologna
- [2] Mauro Bertolini “Corso di aggiornamento in navigazione moderna”
- [3] Jean-Marie Zogg “GPS Basics Introduction to the system Application overview”
- [4] www.tanzilli.com “Corso PIC by Example”
- [5] www.xilinx.com