

Università degli studi di Pisa
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea Specialistica in Scienze Fisiche

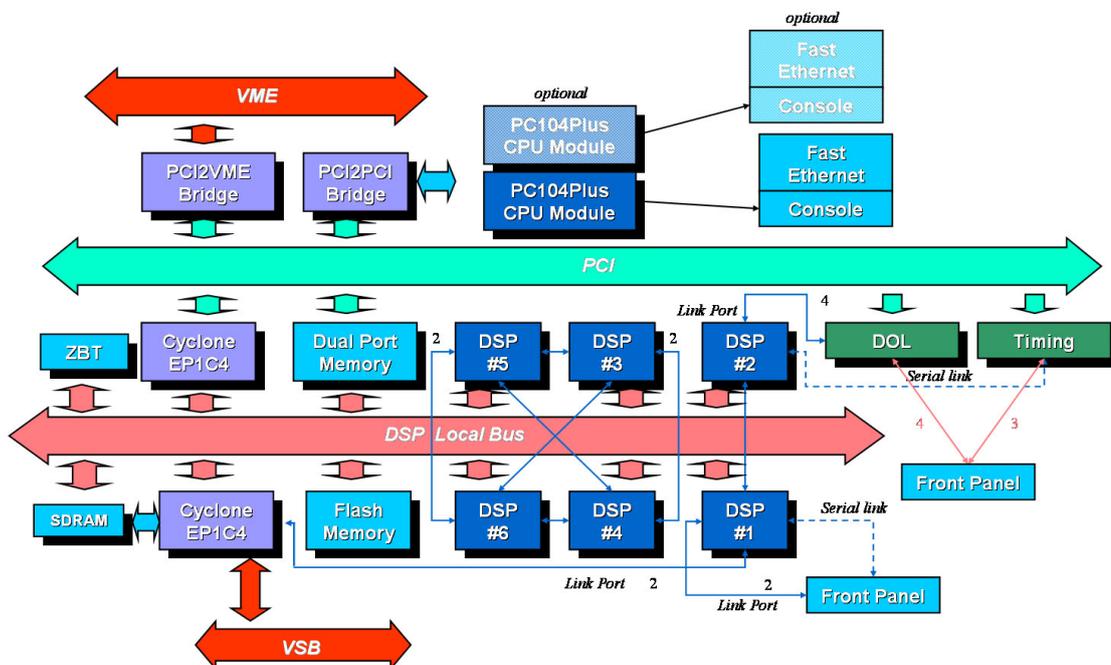
Anno Accademico 2004-05

Tesi di Laurea Specialistica

Candidato: Valerio Boschi

Relatori: Prof. Francesco Fidecaro, Prof. Diego Passuello

Sviluppo di un circuito per l'elaborazione digitale dei segnali dedicati al controllo delle sospensioni di Virgo



Indice

1	Le onde gravitazionali	3
1.1	Origine fisica	3
1.2	Generazione di onde gravitazionali	5
1.3	Sorgenti astrofisiche	7
1.3.1	Binarie coalescenti	7
1.3.2	Supernove	8
1.3.3	Fondo stocastico	9
1.4	Rivelazione interferometrica	10
2	Il progetto Virgo	13
2.1	Le cavità Fabry-Perot	13
2.2	Lo schema ottico	14
2.3	Il locking e il controllo globale	15
2.4	Il superattenuatore	17
2.4.1	I filtri passivi	18
2.4.2	Lo stadio di pre-isolamento	19
2.4.3	Lo stadio finale	20
2.5	Il controllo locale	20
2.5.1	Controllo del pendolo invertito	22
2.5.2	Sensori e attuatori	22
2.6	Il sistema a vuoto	25
2.7	Curva di sensibilità	26
3	Controlli digitali nelle sospensioni di Virgo	29
3.1	Controlli in retroazione	29
3.1.1	Trasformata di Laplace	30
3.1.2	Sistemi di controllo a tempo continuo	32
3.1.3	Trasformata Z	36
3.1.4	Sistemi di controllo a tempo discreto	37
3.2	Filtri digitali in Virgo	39
3.2.1	Trasformazione bilineare	39
3.2.2	Precisione numerica	41
3.2.3	Rappresentazione numerica	43

3.2.4	Forme di implementazione	46
3.2.5	Filtri del I ordine	47
3.2.6	Filtri del II ordine	48
3.2.7	Filtri del II ordine a bassa frequenza	51
3.3	L'attuale sistema di controllo	53
3.3.1	Scheda ADC816b	53
3.3.2	Scheda DAC820b	55
3.3.3	Scheda DSP	55
3.3.4	RIO8062	59
3.3.5	<i>Digital Optical Link</i> (DOL)	59
3.3.6	<i>Timing</i>	59
3.3.7	Sincronizzazione	60
3.3.8	Limiti del sistema	63
4	I DSP	65
4.1	Elaborazione in tempo reale	65
4.2	Architettura dei processori	66
4.2.1	La famiglia Intel x86	66
4.2.2	I RISC	67
4.2.3	Architetture Harvard e SHARC	68
4.3	L'ADSP21160N	70
4.3.1	Bus interni	70
4.3.2	Unità di elaborazione	71
4.3.3	Sequenziatore di programma	74
4.3.4	DAG1 e DAG2	76
4.3.5	Memoria	76
4.3.6	<i>Link ports</i> e configurazioni multiprocessore	77
4.3.7	Porte seriali	79
5	Progetto del circuito	80
5.1	Schema generale del circuito	80
5.1.1	Unità di elaborazione	84
5.1.2	Le FPGA Cyclone EP1C4	85
5.1.3	Memoria	87
5.1.4	Modulo PC/104	90
5.2	Layout del circuito	91
A	Schemi elettrici	94
	Bibliografia	95
	Ringraziamenti	97

Introduzione

La collaborazione italo-francese VIRGO si propone l'osservazione della debole interazione delle onde gravitazionali con la materia attraverso l'uso di un interferometro di Michelson i cui specchi terminali sono sostituiti da cavità Fabry-Perot di 3 km di lunghezza.

L'eccezionale livello di isolamento sismico richiesto ha reso necessario lo sviluppo di un complesso dispositivo meccanico dedicato: il superattenuatore (SA). Esso, attraverso una serie di filtri passivi in cascata, riesce a limitare efficacemente la densità spettrale dello spostamento residuo dello specchio a circa 10^{-18} m / $\sqrt{\text{Hz}}$ a 4 Hz.

A frequenze inferiori tuttavia intervengono i modi normali del SA stesso causando uno spostamento residuo molto più grande (0.1 mm rms) di quello richiesto (10^{-12} m rms di massima oscillazione dello specchio) per mantenere l'interferometro nel punto di lavoro. Il problema viene risolto usando un sistema di controllo digitale multidimensionale che agisce gerarchicamente sul punto di sospensione del pendolo invertito (damping inerziale), sull'ultimo stadio che sorregge lo specchio, detto marionetta, e sullo specchio stesso attraverso una massa di riferimento. A tal fine sono presenti sul SA sensori di posizione (*LVDI*), di accelerazione ed attuatori elettromagnetici.

I segnali provenienti dai sensori, quantizzati a 16 bit, giungono ad un circuito di elaborazione digitale nel quale, al fine di smorzare le risonanze presenti nella funzione di trasferimento del SA, si implementano filtri numerici in cascata. Le sequenze filtrate, convertite in segnali analogici da un DAC a 20 bit, vengono quindi inviate agli attuatori che operano fisicamente le correzioni sul SA. Le caratteristiche dell'unità di elaborazione presente nel circuito hanno conseguenze decisive sulle prestazioni del sistema di controllo: in particolare, per un filtro digitale, le risoluzioni con cui è possibile posizionare un polo o uno zero e con cui è possibile definire il relativo fattore di qualità, crescono linearmente con la precisione numerica disponibile.

Il processore attualmente usato nel controllo locale di Virgo è il Motorola DSP96002. Esso appartiene ad una vasta classe di elaboratori, detti *DSP*, che, grazie ad un'architettura con spazi di memoria separati per istruzioni e dati e ad un'unità aritmetica non *pipe-lined*, sono ottimizzati per l'esecuzione di algoritmi di elaborazione digitale del segnale. Oltre a possedere prestazioni più elevate nell'elaborazione numerica rispetto ai processori tradizionali,

i DSP consentono una completa prevedibilità dei tempi di calcolo. Questa è un requisito fondamentale per un sistema in *tempo reale* qual è il controllo del SA. La violazione dei vincoli temporali presenti nell'interferometro può avere conseguenze critiche sulla stabilità del sistema compromettendo il funzionamento dell'intero strumento.

Sebbene il sistema attuale consenta di ottenere notevoli livelli di attenuazione (da 10^{-4} m a 10^{-7} m rms di spostamento dello specchio a 0.3 Hz), la crescita della potenza di calcolo dei DSP nell'arco di quasi 10 anni dal progetto originario dà l'opportunità di migliorare ulteriormente le prestazioni del controllo ed estenderne le applicazioni.

L'oggetto di questa tesi è lo sviluppo di un circuito per l'elaborazione digitale del segnale che costituisca l'evoluzione di quello attualmente usato in Virgo. La progettazione elettronica è stata effettuata facendo uso di software *CAD*. Il processore scelto è l'ADSP21160N della Analog Devices. Esso appartiene alla famiglia dei DSP SHARC e possiede tre bus e tre unità logiche separate per la gestione di istruzioni, dati e I/O. Vi sono inoltre 2 unità di calcolo che consentono al processore di eseguire numerosi algoritmi tipici dell'elaborazione del segnale in modalità SIMD (Singola Istruzione Molteplici Dati). In tale modalità entrambe le unità di calcolo eseguono la stessa istruzione ma su dati differenti incrementando così le prestazioni. A differenza del sistema attuale, è stata scelta una configurazione multiprocessore che fa uso di 6 DSP. La potenza di calcolo complessiva prevista è di 3.4 GigaFLOPS di picco da confrontare con gli attuali 90 MegaFLOPS. Oltre alle unità di elaborazione sono previsti 512 MB di memoria dinamica (*SDRAM*), 4 MB di memoria statica (*ZBT*), e 256k x 36 bit di memoria Dual-Port. E' inoltre previsto l'uso di due logiche programmabili (*FPGA*) che consentono al circuito di gestire le memorie e di interfacciarsi con il resto dell'elettronica dell'esperimento, attraverso i bus usati in Virgo (VME, VBeX, VSB), e con un qualsiasi PC, attraverso il comune bus PCI.

La notevole potenza di calcolo ottenibile dal sistema progettato potrebbe consentire di estendere il campo di applicazioni del controllo del SA anche ad altri sottosistemi dell'interferometro e all'analisi dei dati prodotti dall'esperimento. Si potrebbero inoltre sviluppare algoritmi più elaborati per il controllo, usando ad esempio metodi multivariabile (filtri di Kalman, tecnica LQG), incrementandone così le prestazioni.

Capitolo 1

Le onde gravitazionali

1.1 Origine fisica

Una delle espressioni basilari della relatività ristretta è quella di intervallo tra due eventi infinitamente vicini nello spazio-tempo[1]:

$$ds^2 = c^2 dt^2 - dx^2 - dy^2 - dz^2 \quad (1.1)$$

Esso può essere scritto in notazione tensoriale come

$$ds^2 = \eta_{\mu\nu} dx^\mu dx^\nu \quad (1.2)$$

dove $\eta_{\mu\nu} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$ è la metrica di Minkowski. Nella rela-

tività generale lo spazio non è necessariamente piatto e la definizione di intervallo diventa:

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu \quad (1.3)$$

dove $g_{\mu\nu}$ non è più un tensore costante. Se la curvatura è piccola può essere fatta un'approssimazione lineare della metrica:

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} \quad (1.4)$$

dove tutti gli elementi del tensore $h_{\mu\nu}$ sono molto minori di 1.

La grandezza che descrive la curvatura dello spazio tempo indipendentemente dal sistema di riferimento è un tensore del 4° ordine, detto tensore di Riemann, dato da

$$R_{\nu\rho\sigma}^\mu = \partial_\rho \Gamma_{\nu\sigma}^\mu - \partial_\sigma \Gamma_{\nu\rho}^\mu + \Gamma_{\alpha\rho}^\mu \Gamma_{\nu\sigma}^\alpha - \Gamma_{\alpha\sigma}^\mu \Gamma_{\nu\rho}^\alpha \quad (1.5)$$

dove $\Gamma_{\nu\rho}^\mu$ sono i simboli di Christoffel, funzioni del tensore metrico:

$$\Gamma_{\nu\rho}^\mu = \frac{1}{2} g^{\mu\sigma} (\partial_\rho g_{\sigma\nu} + \partial_\nu g_{\sigma\rho} - \partial_\sigma g_{\nu\rho}) \quad (1.6)$$

La contrazione del tensore di Riemann è un tensore del 2° ordine detto tensore di Ricci dato da

$$R_{\mu\nu} = g^{\rho\sigma} R_{\rho\mu\sigma\nu} = \partial_\rho \Gamma_{\mu\nu}^\rho - \partial_\nu \Gamma_{\mu\rho}^\rho + \Gamma_{\mu\nu}^\rho \Gamma_{\rho\sigma}^\sigma - \Gamma_{\mu\rho}^\sigma \Gamma_{\nu\sigma}^\rho \quad (1.7)$$

L'equazione che descrive il campo gravitazionale in relatività generale, detta equazione di Einstein risulta

$$R_{\mu\nu} = \frac{8\pi G}{c^4} (T_{\mu\nu} - \frac{1}{2} g_{\mu\nu} T_\lambda^\lambda) \quad (1.8)$$

dove $T_{\mu\nu}$ è il tensore energia-impulso che descrive la densità e il flusso di energia e di impulso del sistema. Usando la (1.4) è possibile sviluppare al primo ordine sia il tensore di Ricci che i simboli di Christoffel ottenendo

$$R_{\mu\nu} \simeq \partial_\nu \Gamma_{\lambda\mu}^\lambda - \partial_\lambda \Gamma_{\mu\nu}^\lambda + o(h^2) \quad (1.9)$$

e

$$\Gamma_{\mu\nu}^\lambda \simeq \frac{1}{2} \eta^{\lambda\rho} (\partial_\mu h_{\rho\nu} + \partial_\nu h_{\rho\mu} - \partial_\rho h_{\mu\nu}) + o(h^2) \quad (1.10)$$

Al fine di semplificare ulteriormente i calcoli, è necessario introdurre una particolare scelta di coordinate, detta gauge armonica, data da

$$g^{\mu\nu} \Gamma_{\mu\nu}^\lambda = 0 \quad (1.11)$$

Usando la (1.10), questa espressione può essere scritta come al primo ordine come

$$\partial_\mu h_\nu^\mu = \frac{1}{2} \partial_\nu h_\mu^\mu \quad (1.12)$$

Quindi, usando la (1.9) e la (1.12), l'equazione di Einstein linearizzata in h assume la forma:

$$\square h_{\mu\nu} = \frac{16\pi G}{c^4} (T_{\mu\nu} - \frac{1}{2} g_{\mu\nu} T_\lambda^\lambda) \quad (1.13)$$

dove è stato introdotto l'operatore $\square = \frac{1}{c} \frac{\partial^2}{\partial t^2} - \nabla^2$. L'insieme delle soluzioni sarà dato dalla somma di una soluzione particolare della (1.13) con le soluzioni dell'omogenea associata. Queste ultime altro non sono che una sovrapposizione di onde che si propagano con velocità c date da

$$h_{\mu\nu}(x) = e_{\mu\nu} \exp(ik_\lambda x^\lambda) + e_{\mu\nu}^* \exp(-ik_\lambda x^\lambda) \quad (1.14)$$

dove k_λ è il 4-vettore d'onda ed $e_{\mu\nu}$ è una matrice simmetrica detta tensore di polarizzazione. Dei 10 gradi di libertà di $e_{\mu\nu}$ è possibile eliminarne 4 usando la condizione (1.12). Introducendo un'ulteriore scelta di coordinate detta gauge trasversa a traccia nulla (*TT-gauge*), il tensore di polarizzazione assume la forma

$$e_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & a & b & 0 \\ 0 & b & -a & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (1.15)$$

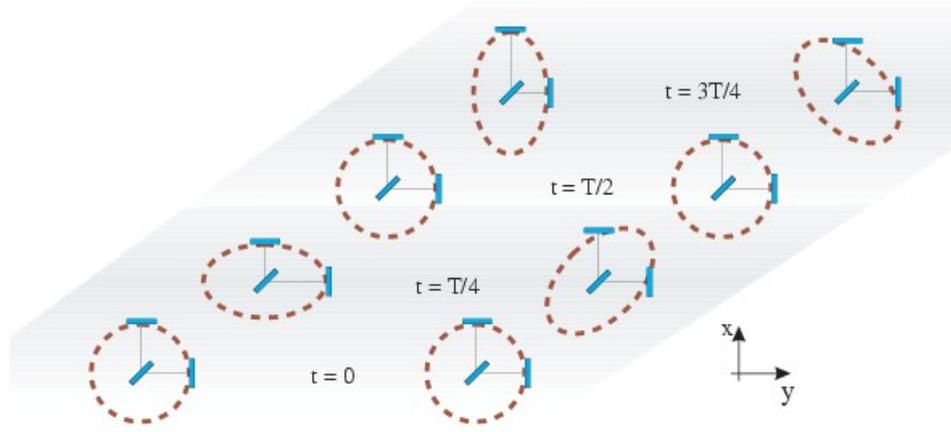


Figura 1.1: Effetto delle due diverse polarizzazioni di un'onda gravitazionale su un insieme di masse in caduta libera disposte su due circonferenze.

Quindi $e_{\mu\nu}$ può essere scritto come somma di due componenti

$$e_{\mu\nu} = a(e_+)_{\mu\nu} + b(e_\times)_{\mu\nu} \quad (1.16)$$

$$\text{con } (e_+)_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \text{ ed } (e_\times)_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Le matrici $(e_+)_{\mu\nu}$ ed $(e_\times)_{\mu\nu}$ sono gli unici due gradi di libertà fisici delle onde gravitazionali e rappresentano le due direzioni ortogonali di polarizzazione per un'onda gravitazionale che si propaga lungo l'asse \hat{z} . Nella fig. 1.1 è mostrato come le due componenti interagiscono con insiemi di masse in caduta libera. L'effetto di $(e_+)_{\mu\nu}$ è di allungare le distanze lungo l'asse \hat{x} e di contrarle lungo \hat{y} , quello di $(e_\times)_{\mu\nu}$ è analogo ma i suoi assi principali sono ruotati di 45° .

1.2 Generazione di onde gravitazionali

L'analogia formale tra l'equazione di Coulomb e quella di Newton suggerisce la possibilità di sviluppare, come in elettrodinamica, la radiazione gravitazionale con un'espansione in multipoli. Si può ad esempio definire un momento di dipolo gravitazionale [2]:

$$d_g = \int dV \rho(r) r \quad (1.17)$$

dove $\rho(r)$ è la densità di massa ed r la distanza dall'origine. Tuttavia l'equivalenza tra massa inerziale e massa gravitazionale richiede il rispetto delle leggi di conservazione della meccanica classica. Poiché \dot{d}_g rappresenta la quantità di moto, grandezza che rimane costante per un sistema isolato, non vi può essere radiazione associata a questo momento di distribuzione.

Il primo termine in grado di generare onde gravitazionali è quello di quadrupolo che, nel caso di moti interni alla sorgente lenti rispetto a c , può essere scritto:

$$I_{\mu\nu} \equiv \int dV \left(x_\mu x_\nu - \frac{1}{3} \delta_{\mu\nu} r^2 \right) \rho(r) \quad (1.18)$$

Il campo associato a questa componente dominante è

$$h_{\mu\nu} = \frac{2G}{Rc^4} \ddot{I}_{\mu\nu} \quad (1.19)$$

Per avere una stima preliminare, usiamo queste relazioni nel caso di una stella binaria, una delle più comuni sorgenti di onde gravitazionali. Consideriamo il moto circolare con frequenza f_{orb} , la massa M uguale per entrambi gli oggetti e la distanza relativa pari a $2r_0$. Applicando successivamente (1.18) e (1.19) otteniamo

$$h_{xx} = -h_{yy} = \frac{32\pi^2 G}{Rc^4} M r_0^2 f_{orb}^2 \cos [2 (2\pi f_{orb}) t] \quad (1.20)$$

$$h_{xy} = h_{yx} = \frac{-32\pi^2 G}{Rc^4} M r_0^2 f_{orb}^2 \sin [2 (2\pi f_{orb}) t] \quad (1.21)$$

Sapendo che l'ampiezza di un'onda gravitazionale è una quantità adimensionale ed $f_{orb} = \frac{GM}{16\pi^2 r_0^3}$, è possibile riordinare la (1.20) e la (1.21) in modo da ottenere un'espressione approssimata:

$$|h| \simeq \frac{r_{s1} r_{s2}}{r_0 R} \quad (1.22)$$

dove $r_{s1} = r_{s2} = \frac{2GM}{c^2}$ sono i raggi di Schwarzschild delle due stelle. Inserendo possibili valori sperimentali per una binaria costituita da stelle di neutroni ($M \simeq 1.4M_\odot$, $f_{orb} \simeq 400$ Hz, $r_0 = 20$ Km), collocata nell'ammasso di galassie più vicino alla Terra (Virgo, $R \simeq 15$ Mpc), otteniamo

$$|h| \simeq 1 \cdot 10^{-21} \quad (1.23)$$

L'energia persa a causa dell'emissione di onde gravitazionali è stata misurata per la prima volta nel caso della pulsar binaria PSR1913+16. Il periodo orbitale di questo sistema, scoperto nel 1974 da R.Hulse e J.Taylor, è calato costantemente nell'arco di 15 anni, riducendosi complessivamente di circa 10 s. Questo fenomeno, osservato anche su altri sistemi, è spiegabile solo con l'emissione di radiazione gravitazionale e rappresenta attualmente l'unica prova indiretta della sua esistenza.

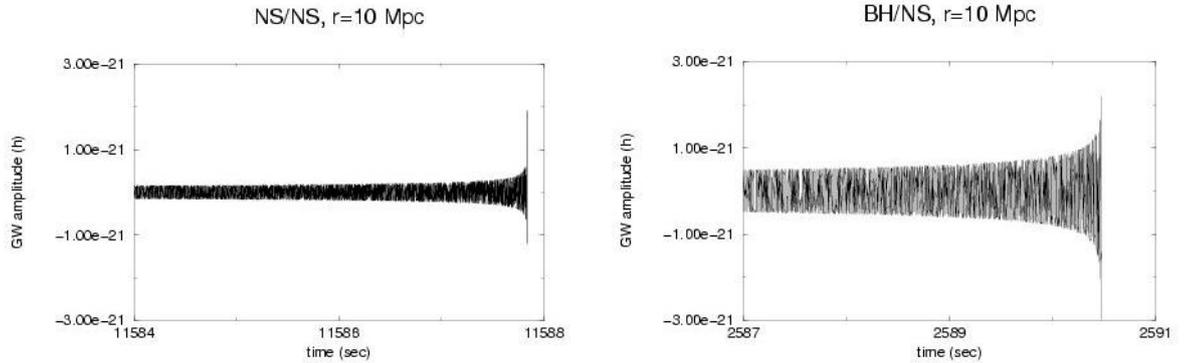


Figura 1.2: *Forme d'onda teoriche della componente h_+ della radiazione gravitazionale emessa dai due tipi principali di binarie coalescenti (NS/NS e BH/NS), poste ad una distanza $r = 10$ Mpc, nella fase finale della loro evoluzione.*

1.3 Sorgenti astrofisiche

Si possono distinguere tre tipi di sorgenti di onde gravitazionali in base al comportamento nel dominio del tempo della radiazione emessa:

- sorgenti periodiche o quasi-periodiche (binarie coalescenti, stelle di neutroni rotanti)
- sorgenti impulsive (supernove)
- sorgenti stocastiche (radiazione gravitazionale di fondo)

1.3.1 Binarie coalescenti

Sono sistemi binari costituiti da oggetti molto compatti, come stelle di neutroni o buchi neri, che perdono energia emettendo radiazione gravitazionale. Il loro periodo orbitale decresce costantemente e le due stelle finiscono per spiralizzare una su l'altra dopo un tempo τ_c detto tempo di coalescenza. Tra tutte le sorgenti di onde gravitazionali le binarie coalescenti sono quelle in cui sono riposte le maggiori speranze da parte della comunità scientifica. Da un lato infatti i sistemi noti di questo tipo, come la già citata PSR1913+16 o la PSR1534+12, sembrano rispettare le previsioni teoriche, dall'altro l'intensità attesa della loro radiazione gravitazionale risulta entro la sensibilità degli interferometri terrestri.

Si possono effettuare calcoli al primo ordine sulle grandezze caratteristiche di questi sistemi usando l'approssimazione *newtoniana/quadrupolare*. Essa consiste nel considerare le masse puntiformi, trascurando le deformazioni mareali, e nel limitare l'emissione gravitazionale al momento di quadrupolo. In questo modo si ottiene la legge che descrive l'andamento della

frequenza orbitale f_{orb} in funzione del tempo:

$$f_{orb} = \left(f(0)^{-8/3} - \pi^{8/3} \frac{256}{5} \frac{G^{5/3}}{c^5} M^{5/3} t \right)^{-3/8} \quad (1.24)$$

dove $M = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}$ con m_1 ed m_2 le masse delle due stelle. E' poi possibile ricavare il tempo di coalescenza

$$\tau_c = 3.0 \frac{f_{orb}(0)}{100\text{Hz}} \left(\frac{M}{M_\odot} \right)^{-5/3} \quad (1.25)$$

dove M_\odot indica la massa del sole. Infine le ampiezze delle due componenti di polarizzazione di h valgono:

$$h_+(r, \theta, t) = 6.0 \cdot 10^{-21} (1 + \cos^2 \theta) \left(\frac{M}{M_\odot} \right)^{5/3} \left(\frac{f_{orb}(t)}{1\text{kHz}} \right)^{2/3} \left(\frac{1\text{Mpc}}{R} \right) \cos(2\pi f_{orb} t)$$

$$h_\times(r, \theta, t) = \pm 1.2 \cdot 10^{-20} \cos \theta \left(\frac{M}{M_\odot} \right)^{5/3} \left(\frac{f_{orb}(t)}{1\text{kHz}} \right)^{2/3} \left(\frac{1\text{Mpc}}{R} \right) \sin(2\pi f_{orb} t)$$

dove R è la distanza del sistema. In figura 1.2 sono mostrate le forme d'onda teoriche associate alla componente h_+ per binarie coalescenti, costituite da due stelle di neutroni (NS/NS) con $m_1 = m_2 = 1.4 M_\odot$, o da un buco nero e una stella di neutroni (BH/NS) con $m_1 = 1.4 M_\odot$ e $m_2 = 10 M_\odot$, nella fase finale della loro evoluzione. Come mostrano i due grafici, nell'ultima fase della coalescenza, l'onda assume una forma tipica, detta *chirp*, caratterizzata da un'oscillazione di ampiezza e frequenza crescente.

Sebbene i meccanismi alla base delle binarie coalescenti siano ben compresi, vi sono notevoli incertezze sul rate degli eventi osservabili. Le stime variano in un range compreso tra $1/10^5$ anni e $1/10^6$ anni per ogni galassia. Per avere un rate di alcuni eventi all'anno è necessario monitorare una distanza di circa 400 Mpc.

1.3.2 Supernove

L'esplosione di una supernova avviene, come noto, quando una stella massiva, terminato il proprio combustibile nucleare, non riesce più a sostenere la propria gravità. Conseguentemente il nucleo centrale della stella inizia a collassare generando una gigantesca onda d'urto che causa a sua volta l'espulsione dello strato più esterno della supernova. Mentre l'emissione elettromagnetica, prodotta in questi eventi, è associata all'espulsione del mantello esterno, la radiazione gravitazionale è legata al collasso del nucleo.

Sebbene alcuni anni fa le supernove fossero ritenute le sorgenti di onde gravitazionali più facilmente rivelabili, è oggi noto che l'intensità della

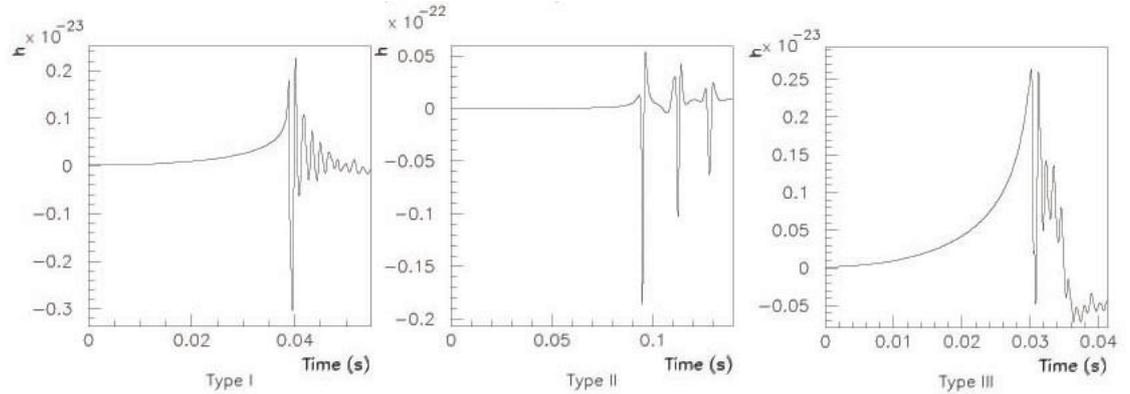


Figura 1.3: Esempi di forme d'onda teoriche delle onde gravitazionali emesse dai tre tipi principali di supernove, poste ad una distanza di 10 Mpc, durante la fase di collasso del loro nucleo.

radiazione emessa durante il collasso del nucleo può variare notevolmente a seconda della percentuale di massa stellare convertita in radiazione e del tipo di simmetria associata al collasso stesso. In particolare una perfetta sfericità del collasso non comporta alcuna emissione di radiazione gravitazionale. Una stima dell'ampiezza di h è fornita dalla relazione:

$$h_+ \sim h_\times \sim C \frac{10\text{Mpc}}{R} \quad (1.26)$$

dove C è compreso tra 10^{-21} e 10^{-24} a seconda dei processi fisici responsabili dell'emissione di onde gravitazionali durante il collasso. La frequenza attesa è dell'ordine del kHz. La fig. 1.3 mostra tre forme d'onda teoriche tratte da un catalogo, ottenuto attraverso una simulazione numerica effettuata nel 1997 da T.Zwinger ed E.Muller [3], costituito da 78 segnali corrispondenti a diverse condizioni iniziali per la rotazione e la distribuzione del momento angolare della stella.

Il rate previsto per questi eventi è compreso tra 1/30 anni e 1/40 anni per galassia. L'osservazione dell'ammasso Virgo, distante 20 Mpc e costituito da circa 2500 galassie dovrebbe quindi fornire un rate di alcuni eventi all'anno.

1.3.3 Fondo stocastico

Oltre ad eventi impulsivi, periodici e quasi-periodici è possibile considerare un segnale stocastico di fondo. Un segnale stocastico è caratterizzato da uno spettro continuo e un'ampiezza pressoché costante. Una possibile origine di tale fondo è data dalla sovrapposizione, nel dominio del tempo o della frequenza, di segnali provenienti da molte sorgenti. E' stato previsto ad esempio che il collasso di vaste popolazioni di buchi neri potrebbe generare,

a seconda della loro distribuzione in massa e in tempo di formazione, segnali di intensità e frequenza sufficienti per essere rilevati dagli attuali esperimenti.

Una sorgente stocastica di particolare interesse è costituita dall'equivalente gravitazionale della ben nota radiazione cosmica di fondo a 3 K. È stato infatti ipotizzato che fluttuazioni, presenti nell'universo primordiale, del campo associato ai gravitoni - particelle che molte teorie quantistiche della gravitazione prevedono costituire i bosoni vettori dell'interazione gravitazionale - siano state amplificate durante l'*inflazione*, una fase di rapida espansione dell'universo presente in molti modelli cosmologici. A seconda dei valori dei parametri che caratterizzano l'inflazione, l'intensità della radiazione gravitazionale di fondo potrebbe raggiungere livelli rivelabili.

1.4 Rivelazione interferometrica

Come scrive R.Dicke in uno dei primi lavori sulla relatività sperimentale [4]: “ In altri campi della fisica lo sperimentatore si trova di fronte al problema di scegliere il più importante di un gran numero di esperimenti possibili. Con la gravitazione il problema è diverso. Ci sono così pochi esperimenti possibili e di tale importanza che ognuno di essi deve essere eseguito ”.

Storicamente i primi esperimenti per la rivelazione delle onde gravitazionali sono quelli condotti negli anni '60 da J.Weber con due barre risonanti [5]. In questi strumenti, che in versioni molto perfezionate vengono usati ancora oggi ¹, viene misurata l'oscillazione indotta dalle perturbazioni della metrica in una massa metallica di forma cilindrica, raffreddata ad una temperatura vicina allo zero assoluto.

Oggi, tuttavia, le maggiori attenzioni della comunità scientifica sono rivolte verso un altro tipo di strumento: l'interferometro di Michelson, lo stesso dispositivo che ha dimostrato uno dei principi fondamentali della relatività ristretta, l'assenza dell'etere. Per capire perché viene usato consideriamo un interferometro con i bracci allineati lungo gli assi \hat{x} e \hat{y} . Esaminiamo il braccio \hat{x} . Partiamo dal fatto che la luce connette eventi nello spazio-tempo separati da un intervallo nullo $ds^2 = 0$. Quindi ponendo la (1.2) uguale a zero otteniamo

$$ds^2 = -c^2 dt^2 + \left(1 + h_{11} \left(2\pi ft - \vec{k} \cdot \vec{x}\right)\right) dx^2 = 0 \quad (1.27)$$

Possiamo valutare il tempo che impiega la luce ad andare dal *beam splitter* alla fine del braccio \hat{x} integrando la radice della precedente espressione:

$$\tau_x^{(1)} = \frac{1}{c} \int_0^L \sqrt{1 + h_{11}} dx \approx \frac{1}{c} \int_0^L \left(1 + \frac{h_{11}}{2} \left(2\pi ft - \vec{k} \cdot \vec{x}\right)\right) dx \quad (1.28)$$

¹Gli esperimenti AURIGA, NAUTILUS, EXPLORER, ALTAIR (Italia), ALLEGRO (Louisiana, USA), Niobe (Australia)

dove la radice è stata sviluppata in serie al primo ordine dato che $h \ll 1$. Analogamente possiamo scrivere il tempo per compiere il tragitto inverso:

$$\tau_x^{(2)} = -\frac{1}{c} \int_L^0 \left(1 + \frac{h_{11}}{2} (2\pi ft - \vec{k} \cdot \vec{x}) \right) dx \quad (1.29)$$

Il tempo totale $\tau_x = \tau_x^{(1)} + \tau_x^{(2)}$ è quindi

$$\tau = \frac{2L}{c} + \frac{1}{2c} \int_0^L h_{11} (2\pi ft - \vec{k} \cdot \vec{x}) dx - \frac{1}{2c} \int_L^0 h_{11} (2\pi ft - \vec{k} \cdot \vec{x}) dx \quad (1.30)$$

Scegliamo il caso più semplice di un'onda sinusoidale che si propaga lungo l'asse \hat{z} , con frequenza f_{gw} , polarizzazione + ed ampiezza $h_{11} = -h_{22} = h_0$. I due integrali della (1.30) possono essere facilmente calcolati dopo aver posto $h(t) = h_0 \text{Re}[\exp(i2\pi f_{gw}t)]$. Il tempo totale sarà

$$\tau_x = \tau_0 + \frac{h_0}{4\pi i f_{gw}} \text{Re}[e^{i2\pi f_{gw}\tau_0} - 1] \quad (1.31)$$

dove $\tau_0 = \frac{2L}{c}$ è il tempo classico. Nel caso di un fotone che percorre il braccio \hat{y} le espressioni sono identiche a parte la dipendenza da h_{22} invece che da h_{11} . Quindi

$$\tau_y = \tau_0 - \frac{h_0}{4\pi i f_{gw}} \text{Re}[e^{i2\pi f_{gw}\tau_0} - 1] \quad (1.32)$$

Nel sistema di riferimento del laboratorio questa variazione del tempo di volo rispetto al caso classico può essere interpretata come una alterazione della lunghezza dei due bracci, L_x ed L_y , dell'interferometro:

$$L_x = \frac{1}{2} c\tau_x = L \left(1 + \frac{h(t)}{2} \right) \quad (1.33)$$

$$L_y = \frac{1}{2} c\tau_y = L \left(1 - \frac{h(t)}{2} \right) \quad (1.34)$$

Nel caso di un interferometro lungo 1 Km tale alterazione $L_x - L \sim 10^{-18}$ m. I due fotoni arrivano all'uscita dell'interferometro con una differenza di fase pari a

$$\frac{\delta\phi(t)}{h(t)} = \frac{4\pi L}{\lambda} \text{Re}[e^{i2\pi f_{gw}\tau_0} \text{sinc}(\pi f_{gw}\tau_0)] \quad (1.35)$$

dove $\text{sinc}(x) = \frac{\sin x}{x}$. Nella fig. 1.4 sono riportati i plots di $\frac{\delta\phi(t)}{h(t)}$ in funzione della frequenza per interferometri di diverso tipo e lunghezza. Nella regione piatta a bassa frequenza in cui $f_{gw}\tau_0 \ll 1$ la (1.35) può essere approssimata come

$$\frac{\delta\phi(t)}{h(t)} = \frac{4\pi L}{\lambda} \quad (1.36)$$

formula che mostra esplicitamente come la lunghezza dei bracci sia un parametro decisivo per incrementare la sensibilità dello strumento.

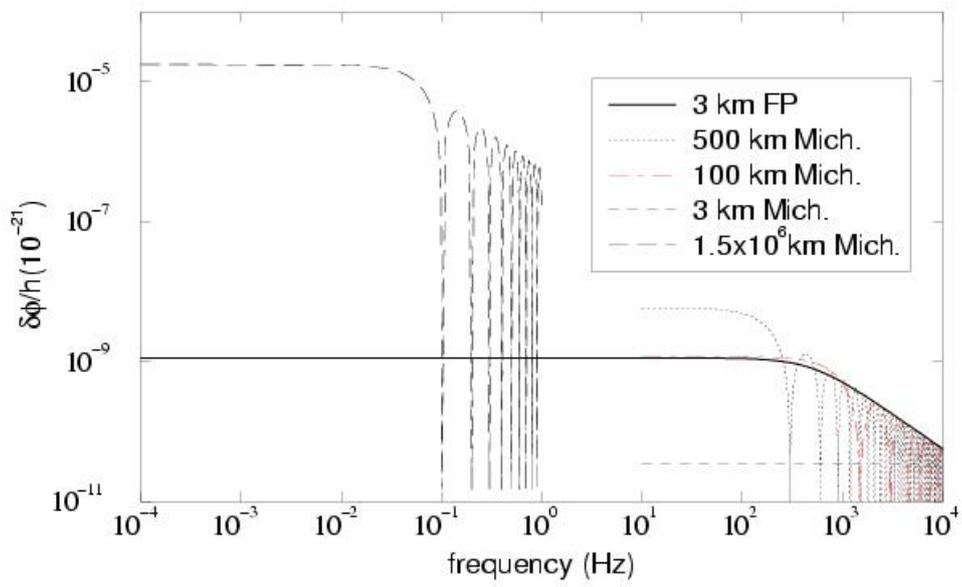


Figura 1.4: Grafici di $\frac{\delta\phi(t)}{h(t)}$ per diverse lunghezze e tipi di interferometri.

Capitolo 2

Il progetto Virgo

La collaborazione italo-francese Virgo è stata fondata nel 1989 con l'obiettivo di realizzare un interferometro per le onde gravitazionali presso Cascina, Pisa. Lo strumento è costituito da un interferometro di Michelson i cui specchi terminali sono sostituiti da cavità Fabry-Perot di 3 Km di lunghezza. La costruzione è ormai ultimata e sono in corso una serie di test che porteranno lo strumento ad una piena operatività entro il 2005.

2.1 Le cavità Fabry-Perot

Come evidenziato nella fig. 1.4 la sensibilità di un interferometro con queste caratteristiche equivale a quella di un Michelson semplice di oltre 100 Km di lunghezza. Le cavità Fabry-Perot consentono infatti di aumentare il cammino ottico dei fotoni di alcuni ordini di grandezza. Senza l'uso delle cavità nessun interferometro terrestre, con sensibilità adeguata a rivelare variazioni della metrica come quelle espresse dalla (1.23), sarebbe economicamente realizzabile. Costituite da due specchi piani posti a distanza L , possono essere considerate l'equivalente ottico delle cavità risonanti usate in elettronica nel campo delle microonde.

La cavità è detta in risonanza quando l'energia immagazzinata è massima, situazione che corrisponde a $2kL = 2n\pi$ dove n è un intero e k il numero d'onda.

Parametro chiave della cavità è la finezza \mathfrak{F} definita come il rapporto tra la larghezza della risonanza (misurata a metà altezza) e l'intervallo tra due risonanze $\Delta f \equiv \frac{c}{2L}$ detto intervallo spettrale libero. Esso dà una misura di quanto vari la fase della luce in uscita in seguito ad una piccola variazione della lunghezza L come quella prodotta da un'onda gravitazionale. In modo particolare nel limite in cui il periodo dell'onda $\frac{1}{f_{gw}} \gg \frac{\mathfrak{F}L}{c}$, una cavità

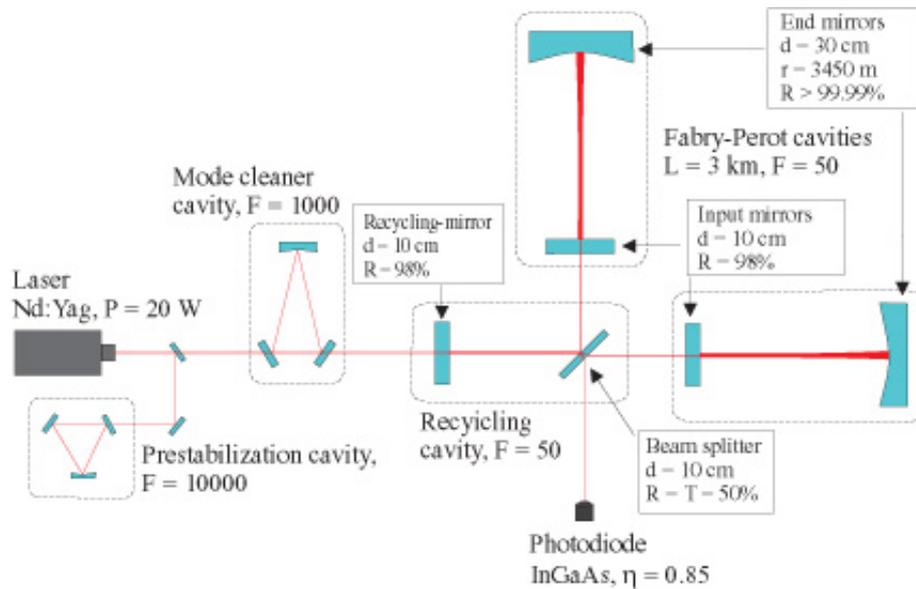


Figura 2.1: Schema ottico di Virgo

Fabry-Perot produce $\frac{2\mathfrak{F}}{\pi}$ volte lo sfasamento ottenuto con un Michelson semplice.

2.2 Lo schema ottico

Lo schema ottico di Virgo è mostrato in fig. 2.1 Costituenti principali sono

- Il laser: è al Nd:Yag con $\lambda = 1064$ nm
- L'injection bench: è costituito dalla cavità di pre-stabilizzazione della frequenza del laser, dagli specchi d'ingresso e di uscita del mode cleaner e dall'ottica necessaria all'allineamento.
- Le cavità Fabry-Perot: sono lunghe 3 Km ed hanno una finezza $\mathfrak{F} = 50$
- Il mode cleaner: è una cavità triangolare lunga circa 150 m con $\mathfrak{F} = 1000$. Due specchi (lo specchio d'ingresso e di uscita) sono montati sull'injection bench, il terzo è separato e connesso all'injection attraverso un condotto dedicato.
- Il power recycling o cavità di ricircolo: la cavità è costituita dallo specchio di ricircolo e dall'interferometro stesso. Si prevede di amplificare la potenza del laser di un fattore 50 (1000 W al *beam splitter*)

- Il detection bench: supporta il mode cleaner di uscita (una piccola cavità triangolare progettata per aumentare il contrasto) e l'ottica necessaria per collimare il fascio sui fotodiodi. All'esterno è presente un banco ottico su cui sono posti i fotodiodi (InGaAs, efficienza quantica $\eta = 0.85$) usati per il locking e la rivelazione del segnale.

2.3 Il locking e il controllo globale

Per locking dell'interferometro si intende il passaggio degli specchi e delle altre componenti dell'interferometro da uno stato libero ad uno stato controllato da un sistema di retroazione in grado di mantenere lo strumento nel punto di lavoro ottimale. Quest'ultimo corrisponde ad una minima intensità luminosa in uscita (frangia scura) e alle cavità Fabry-Perot e ricircolo in risonanza con il laser.

Il sistema di controllo responsabile del locking, detto **controllo globale**, può essere diviso in tre parti:

- il controllo del laser: in frequenza, in potenza e al livello della geometria del fascio
- il controllo longitudinale delle cavità
- il controllo angolare degli specchi.

Il ciclo del controllo funziona a 10 kHz e tale parametro deve essere rigorosamente rispettato al fine di assicurare la stabilità del feedback. Quindi ogni 100 μs il controllo globale deve operare in ordine (vedi fig. 2.2) [6]:

1. la lettura dei fotodiodi
2. la ricostruzione delle lunghezze: attraverso un algoritmo di minimizzazione col metodo del χ^2 che fa uso della matrice ottica del sistema, vengono ricostruite le lunghezze delle quattro cavità.
3. il calcolo delle correzioni: le quattro lunghezze ottenute devono essere tradotte negli spostamenti da applicare agli specchi e nelle correzioni da apportare alla frequenza del laser
4. l'invio delle correzioni: le correzioni calcolate sono inviate alle varie componenti del sistema, laser e superattenuatori, attraverso connessioni in fibra ottica.

I requisiti del locking sono

- *massima oscillazione rms dello specchio*: deve essere inferiore ad 1/10 della larghezza a metà altezza della risonanza della cavità ($\delta L <$

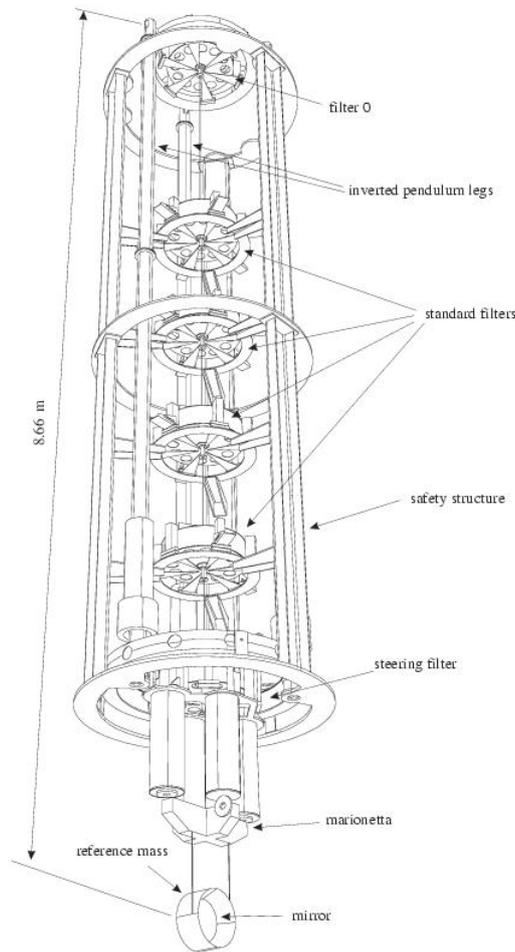


Figura 2.3: Schema di un superattenuatore lungo.

2.4 Il superattenuatore

Sebbene teoricamente solo le componenti delle oscillazioni sismiche dirette lungo il fascio dovrebbero avere effetto sull'interferometro, nel sistema reale tutti i 6 gradi di libertà devono essere attenuati. Vi sono infatti accoppiamenti tra le componenti traslazionali e rotazionali che non possono essere eliminati. Ad esempio la curvatura terrestre genera un angolo tra le verticali degli specchi d'ingresso e di uscita delle cavità Fabry-Perot di $\alpha_{grav} = 3 \cdot 10^{-4}$ rad. Quindi uno spostamento verticale δy si ripercuote lungo la direzione del fascio, producendo una variazione $\alpha_{grav} \delta y$.

Il superattenuatore, mostrato in fig. 2.3, può essere essenzialmente diviso in tre sezioni:

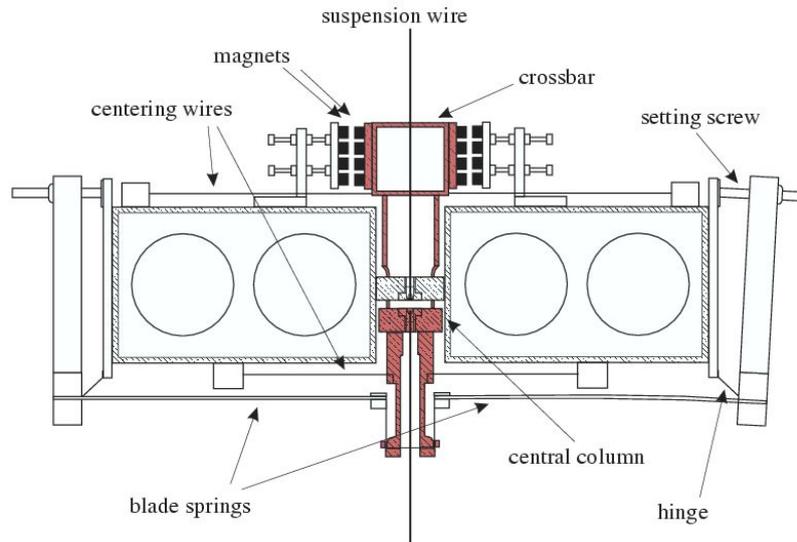


Figura 2.4: Sezione di un filtro passivo standard

- lo stadio di pre-isolamento
- la catena di filtri
- il controllo dello specchio.

L'intera catena, costituita da 6 filtri, è lunga circa 10 m ed ha quindi una frequenza di risonanza fondamentale di $\frac{1}{2\pi} \sqrt{\frac{g}{l}} = 0.16$ Hz. Il primo e l'ultimo stadio detti, *filtro 0* e *filtro 7*, si differenziano dagli altri in quanto sede dei controlli attivi della sospensione.

Non tutte le componenti dell'ottica richiedono la stessa attenuazione sismica: l'injection bench, il detection bench e lo specchio terminale del mode-cleaner sono sospese a dei superattenuatori più piccoli, lunghi circa 2 m, in gergo *torri corte*.

2.4.1 I filtri passivi

Ognuno dei 4 filtri standard è progettato per ridurre la propagazione delle vibrazioni meccaniche di 2 ordini di grandezza per frequenze maggiori di 10 Hz. Il corpo centrale del filtro (vedi fig. 2.4) è costituito da un cilindro rigido di acciaio del diametro di 70 cm, altezza 18.6 cm e peso 100 Kg. La scelta di queste caratteristiche risiede nella necessità di avere un alto momento d'inerzia in modo da abbassare il più possibile le frequenze dei modi rotazionali propri. Il carico degli stadi inferiori viene sostenuto attraverso una serie di lame elastiche triangolari che, agendo come molle, sono responsabili dell'attenuazione verticale. Grazie alla loro forma, la distribuzione

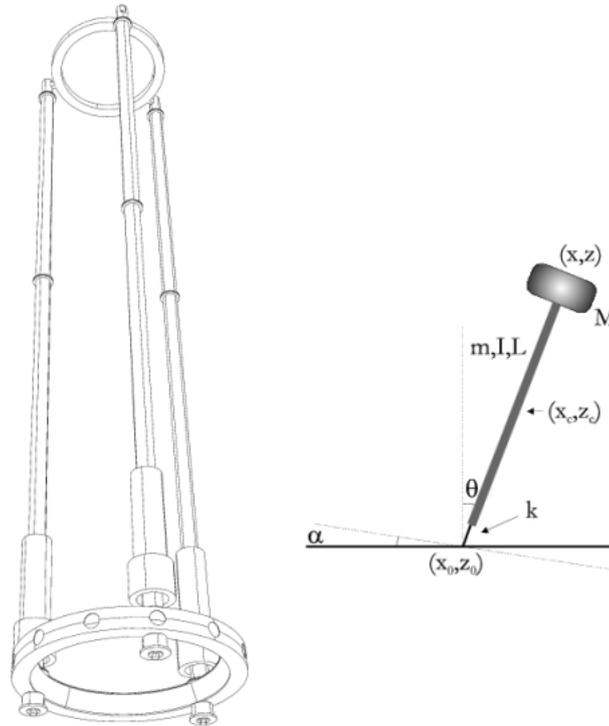


Figura 2.5: a) il pendolo invertito reale b) un semplice modello

dello sforzo, che viene mantenuto a $2/3$ del limite elastico del materiale, è uniforme lungo la lama. Ciascuna lama viene piegata prima di essere montata in modo da assumere una configurazione orizzontale una volta posta sotto sforzo. Inoltre, in seguito ad una serie di studi, è stato individuato il tipo di trattamento più adatto a cui sottoporre il materiale per ridurre i fenomeni di *creep* del metallo [7].

2.4.2 Lo stadio di pre-isolamento

La catena dei filtri passivi è sospesa ad uno stadio dedicato all'attenuazione delle basse frequenze detto stadio di pre-isolamento [8]. Tale stadio è esso stesso un oscillatore lungo \hat{x} , \hat{y} , \hat{z} , $\hat{\theta}_z$ ma con frequenze di risonanza molto inferiori ($f_0 \simeq 30$ mHz) a quelle della catena. Il sistema è costituito da 3 colonne rigide, connesse a terra attraverso giunti flessibili, che supportano un piano detto top stage su cui è posto il cosiddetto *filtro 0*. Le tre gambe possono essere descritte fisicamente come un *pendolo invertito* e sono responsabili dello smorzamento orizzontale. Un modello fisico molto semplice ma utile per comprenderne la dinamica è mostrato in fig. 2.5: una massa

M è sostenuta da una barra rigida di lunghezza l connessa a terra da un giunto flessibile e di rigidità angolare κ . L'equazione del moto è

$$I\ddot{\theta} = -\kappa\theta + Mgl \sin \theta \quad (2.2)$$

dove I è il momento d'inerzia del sistema rispetto al punto di sospensione O e θ l'angolo formato dalla sbarra con la verticale. Nell'approssimazione di piccoli angoli, riscrivendo l'equazione in funzione di $x = \theta l$ e della costante elastica $K = \kappa/l^2$, otteniamo un frequenza di risonanza:

$$f_0 = \frac{1}{2\pi} \sqrt{\frac{K}{M} - \frac{g}{l}} \quad (2.3)$$

Per non introdurre instabilità nel sistema viene scelto un valore di K che renda f_0 piccolo ma non nullo. Il principale vantaggio del pendolo invertito risiede nella facilità con cui può essere controllato. Infatti la forza necessaria per spostare staticamente di un tratto x la massa M è data da $F = M\omega_0^2 x$. E' possibile quindi muovere anche considerevolmente la struttura introducendo rumore limitato, che viene poi filtrato nella banda di rivelazione dalla catena di filtri.

2.4.3 Lo stadio finale

L'ultima sezione ospita il filtro terminale della catena ed una struttura orizzontale che sostiene lo specchio detta *marionetta*. Il filtro 7 è essenzialmente un filtro standard equipaggiato con una serie di dispositivi che gli consentono assolvere funzioni addizionali. Quattro colonne lunghe 1 m si estendono dal filtro supportando ognuna un avvolgimento. A ciascun avvolgimento corrisponde sulla punta dei bracci incrociati che costituiscono la marionetta, un piccolo magnete. In questo modo è possibile esercitare forze e torsioni sullo specchio per correggerne il moto in una banda di 0.1 - 1 Hz.

2.5 Il controllo locale

Il funzionamento passivo del superattenuatore non è sufficiente per i requisiti del locking. In particolare nelle due bande:

- $f < 30 \text{ mHz}$: il rumore sismico è completamente trasmesso alla sospensione e lo spostamento prodotto può arrivare fino a $60 \mu\text{m}$ a causa delle deformazioni mareali della crosta terrestre.
- $30 \text{ mHz} < f < 2 \text{ Hz}$: intervengono i modi normali del superattenuatore stesso causando oscillazioni dello specchio di oltre $10 \mu\text{m rms}$

In entrambi i casi siamo almeno 8 ordini di grandezza oltre i limiti. E' quindi necessario un sistema di controllo che elimini il rumore sismico residuo facendo uso dei sensori e degli attuatori presenti sulla sospensione. Tale controllo

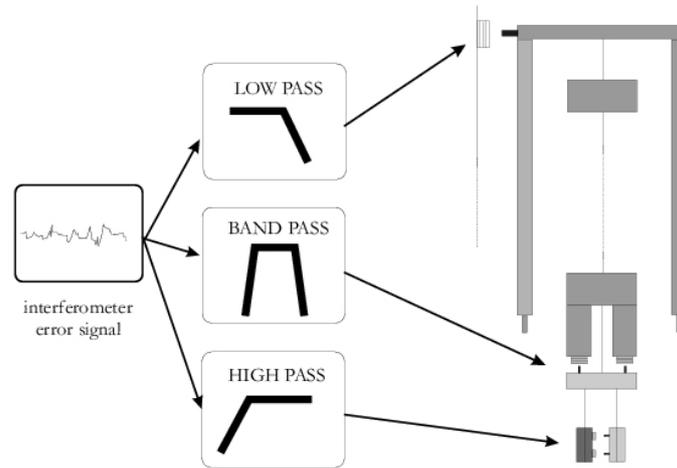


Figura 2.6: *La gerarchia del controllo locale*

non può agire unicamente sullo specchio in quanto l'estensione delle attenuazioni da operare è troppo elevata per la dinamica dell'elettronica digitale usata. Ad esempio il rumore di quantizzazione introdotto dai convertitori digitale-analogico (DAC), per compensare anche solo $1 \mu\text{m}$ di oscillazione dello specchio, una volta iniettato nel range di rivelazione del segnale, avrebbe pesanti effetti sulla sensibilità dell'interferometro. Parametro decisivo è il range dinamico degli attuatori definito come

$$d = \frac{F_{max}}{F_n(f)} [\sqrt{\text{Hz}}] \quad (2.4)$$

dove F_{max} è la forza massima che l'attuatore è in grado di esercitare sul sistema ed $F_n(f)$ è la densità in $\text{N}/\sqrt{\text{Hz}}$ del rumore indotto dall'elettronica.

La soluzione adottata per le sospensioni di Virgo è quella di usare un controllo multidimensionale, detto **controllo locale**, che agisca gerarchicamente, tenendo conto del range dinamico dei vari attuatori, su tre punti del superattenuatore (vedi fig. 2.6):

- il pendolo invertito: grazie alla sua particolare struttura consente spostamenti massimi di $\pm 1 \text{ cm}$ per $f < 0.1 \text{ Hz}$ senza iniettare rumore elettronico.
- la marionetta: su di essa è possibile applicare spostamenti di $\pm 10 \mu\text{m}$ nella banda $0.1 - 1 \text{ Hz}$.
- lo specchio: per $f > 1 \text{ Hz}$ le forze possono essere applicate direttamente alla massa di riferimento posta dietro lo specchio.

L'elettronica del controllo locale verrà discussa nel cap. 3.

2.5.1 Controllo del pendolo invertito

Sul pendolo invertito vi sono due tipi di controllo da operare:

- controllo in posizione: usa gli LVDT per compensare eventuali spostamenti dalla posizione ottimale dello specchio.
- controllo in accelerazione: chiamato anche **damping inerziale**, in quanto usa gli accelerometri come sensori, smorza i modi normali del superattenuatore consentendo il locking dell'interferometro.

Consideriamo ora il pendolo invertito come un sistema lineare che ha come ingresso il vettore $\mathbf{q} = [q_1 q_2 q_3]$ degli attuatori elettromagnetici e come uscita il vettore $\mathbf{u} = [u_1 u_2 u_3]$ dei sensori di posizione o di accelerazione. Un sistema di questo tipo è detto multivariabile o MIMO (Multiple Input Multiple Output), in quanto ha ingressi e uscite multiple. Questo fisicamente significa che, in generale, ogni sensore è in grado di rivelare solo combinazioni di tutti i 3 modi normali del pendolo invertito $(\hat{x}, \hat{y}, \hat{\theta})$ ed ogni attuatore produce movimenti in tutte le 3 direzioni dei modi di oscillazione. Progettare un controllo in retroazione e definirne i criteri di stabilità è molto complesso per un sistema di questo genere. La soluzione adottata da Virgo è quella di operare una trasformazione di coordinate che consenta di passare dallo spazio dei sensori/attuatori reali a quello dei sensori/attuatori virtuali, in cui ogni attuatore eccita un singolo modo normale ed ogni sensore rileva ciascun modo separatamente. Per determinare sperimentalmente questa trasformazione è necessario misurare due matrici \mathcal{S} e \mathcal{D} , dette rispettivamente di *sensing* e di *driving*, definite come:

$$\mathbf{v} = \mathcal{D}\mathbf{q} \quad (2.5)$$

$$\mathbf{x} = \mathcal{S}\mathbf{u} \quad (2.6)$$

dove \mathbf{v} e \mathbf{x} sono gli attuatori e i sensori virtuali. Ottenute \mathcal{S} e \mathcal{D} è possibile spezzare il sistema globale in 3 sistemi singolo ingresso - singola uscita (SISO) ed applicare i metodi del controllo monovariabile.

2.5.2 Sensori e attuatori

Come mostra la fig. 2.7, sul top stage sono montati, ai vertici di un triangolo equilatero, i sensori e gli attuatori necessari per il controllo del pendolo invertito. Vi sono due tipi di sensori

- **LVDT** (Linear Variable Differential Trasformer): sono dispositivi usati per misurare con grande precisione la posizione del pendolo invertito. Funzionano con due insiemi di avvolgimenti: uno, costituito da una bobina singola, viene eccitato con corrente alternata a 20 kHz, l'altro, costituito da 2 bobine, viene usato per misurare il segnale. Quando

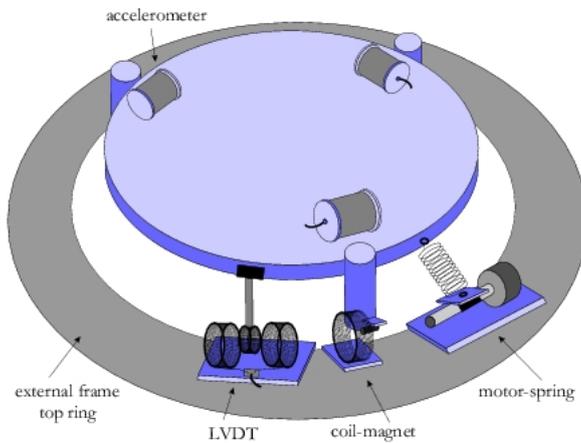


Figura 2.7: Schema dei sensori e degli attuatori montati

l'avvolgimento centrale del sensore si sposta di una quantità δx , sul secondo set si misura un segnale sinusoidale alla stessa frequenza di quello di eccitazione ma con un'ampiezza proporzionale a δx . Una volta calibrati, hanno un comportamento lineare entro l'1% in un range di 1 cm.

- **Accelerometri:** sono stati costruiti espressamente per le sospensioni di Virgo. La struttura meccanica dell'accelerometro è costituita da una sbarra di ottone caricata con due dischi dello stesso materiale è sostenuta da due lamine di CuBe in modo da formare un pendolo invertito [9]. La massa dei due dischi è scelta al fine di ottenere una frequenza di risonanza di circa 5 Hz, incrementando così la sensibilità dello strumento nel range in frequenza dei modi normali del superattenuatore. Un *LVDT* di tipo analogo a quello precedentemente descritto è usato come sensore di posizione delle masse rispetto al punto di sospensione. Lo strumento utilizza un sistema di controllo in retroazione (vedi cap. 3).

Il segnale proveniente dall'*LVDT*, dopo essere stato opportunamente filtrato usando una rete proporzionale-integrale-derivativa (*PID*), viene inviato ad un avvolgimento posto ad un'estremità della sbarra. Esso è sensibile al campo generato da un magnete permanente fissato alla struttura esterna e consente così di mantenere il sistema nella sua posizione di equilibrio agendo come una molla elettronica. Il sistema può essere schematizzato come un oscillatore armonico di massa m , frequenza di risonanza ω_0 e fattore di qualità Q . Nel dominio della frequenza, la forza di richiamo è proporzionale allo spostamento della

massa:

$$\frac{f_{FB}}{m} = -F(\omega)[x(\omega) - x_0(\omega)] \quad (2.7)$$

dove $F(\omega)$ è la funzione di trasferimento della rete di controllo (vedi cap. 3). Combinando la (2.7) con l'equazione del moto otteniamo

$$\frac{f_{FB}}{m} = \frac{H(\omega)F(\omega)}{1 + H(\omega)F(\omega)}[-\omega^2 x_0(\omega)] \quad (2.8)$$

dove

$$H(\omega) = \frac{1}{-\omega^2 + i\omega\omega_0/Q + \omega_0^2}$$

Nelle regioni in frequenza in cui il guadagno ad anello aperto $|H(\omega)F(\omega)|$ è molto maggiore di 1, risulta

$$\frac{f_{FB}}{m} = \omega^2 x_0 \quad (2.9)$$

e la forza di richiamo è quindi proporzionale all'accelerazione.

La sensibilità dello strumento è limitata essenzialmente da due tipi di rumore: il rumore di Johnson dell'avvolgimento secondario dell'*LVDT* (1 nV/ $\sqrt{\text{Hz}}$ a temperatura ambiente) e il rumore dell'amplificatore (0.8 nV/ $\sqrt{\text{Hz}}$). La funzione di sensibilità spettrale, definita come la minima accelerazione che produce in uscita al demodulatore il rumore spettrale misurato, è piatta per frequenze al di sotto della risonanza e vale

$$7 \cdot 10^{-10} \frac{\text{m/s}^2}{\sqrt{\text{Hz}}} \quad (2.10)$$

Questo si traduce in una sensibilità dello strumento agli spostamenti di

$$8 \cdot 10^{-13} \frac{\text{m}}{\sqrt{\text{Hz}}} \quad (2.11)$$

valore oltre 3 ordini di grandezza al di sotto del rumore sismico ($X(f) = 10^{-7}/f^2 \text{ m}/\sqrt{\text{Hz}}$).

Infine l'ampiezza del segnale in uscita può essere scritta come:

$$V_{out} = \sigma(g \sin \theta + \ddot{x}) \quad (2.12)$$

dove σ è un parametro di calibrazione e θ è l'angolo che l'accelerometro, posto sulla piattaforma superiore, forma con l'orizzontale, detto angolo di *tilt*.

Analogamente vi sono due tipi di attuatori, elettromagnetici e meccanici:

- magnete-bobina: sono costituiti da una coppia di bobine e da un magnete centrale di forma cubica di 4 cm di lato ancorato al filtro 0 attraverso una sottile sbarra metallica.

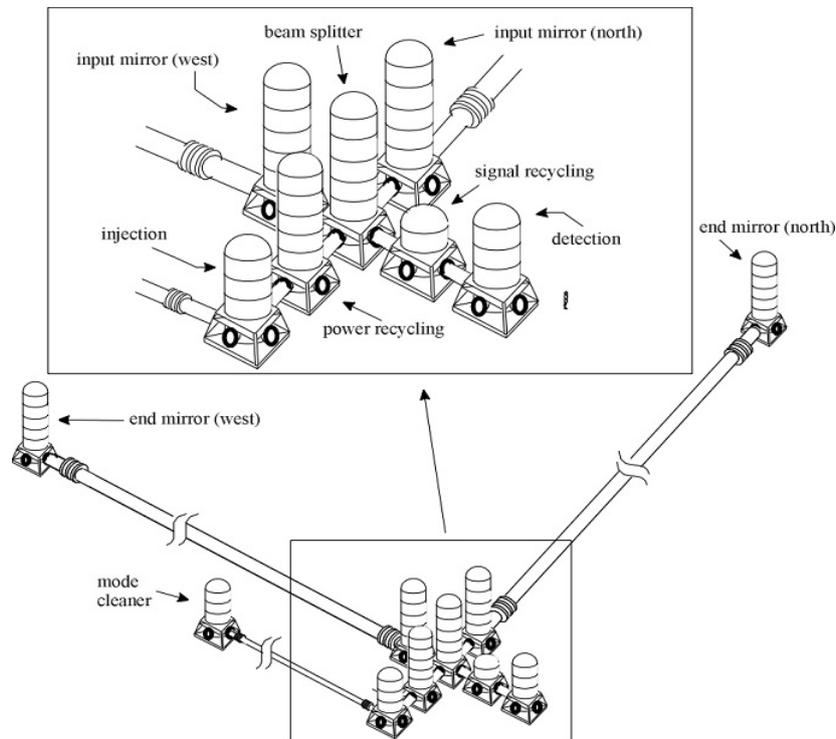


Figura 2.8: *Schema del sistema a vuoto*

- molla-motore: sono costituiti da due molle orizzontali connesse alle gambe del pendolo invertito tramite un supporto controllato attraverso una vite senza fine azionata da un motore elettrico.

2.6 Il sistema a vuoto

Al fine di ridurre le variazioni dell'indice di rifrazione all'interno dei tubi che ospitano il fascio laser, è necessario un sistema ad ultra-alto vuoto che garantisca una pressione parziale di 10^{-9} mbar per l'idrogeno. E' richiesto un limite ancora più forte, 10^{-14} mbar, per gli idrocarburi gassosi datoche tali composti possono fissarsi alla superficie degli specchi degradandone la riflettività. Sebbene anche nelle torri contenenti i superattenuatori venga fatto il vuoto, la presenza dei cavi e dei componenti dell'elettronica di controllo non consente i livelli di ultra-alto vuoto raggiunti nei tubi. Per risolvere il problema ciascuna torre è divisa in due sezioni che comunicano solo attraverso uno stretto condotto in cui scorre il cavo metallico che sostiene l'ultimo stadio del superattenuatore. L'elettronica è posta nella parte superiore che viene mantenuta ad una pressione di 10^{-6} mbar.

La fig. 2.9 mostra il sistema completo: i tubi hanno un diametro di 1.2

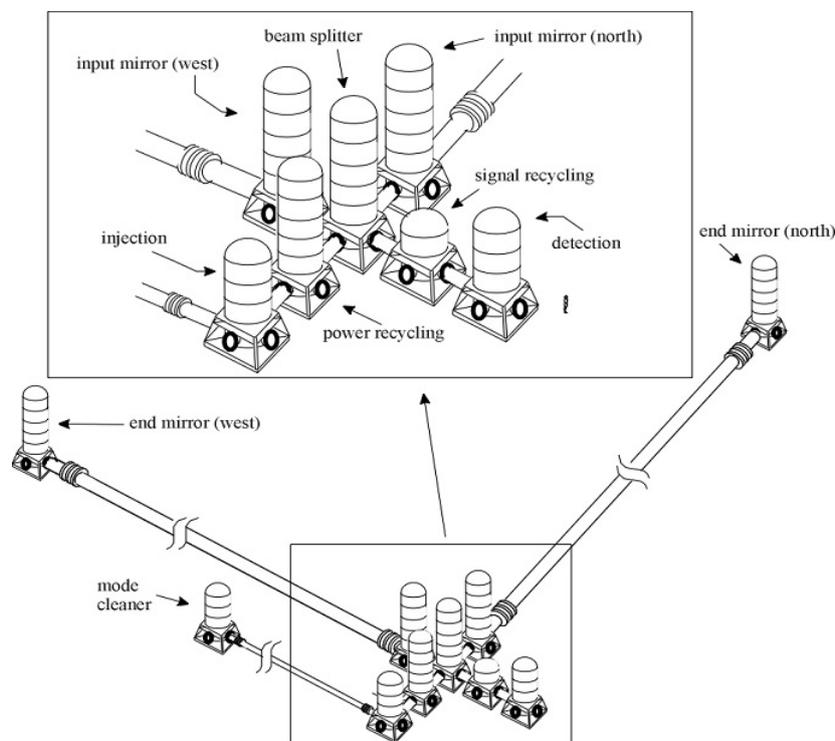


Figura 2.9: Schema del sistema a vuoto

m e sono divisi in sezioni di 15 m ciascuna. Il volume complessivo è di circa 7000 m^3 e rende Virgo il più grande sistema a vuoto d'Europa.

2.7 Curva di sensibilità

La curva di sensibilità di un interferometro, fig. 2.10, riassume le prestazioni dello strumento mostrando i contributi delle varie sorgenti di rumore. Si distinguono

- *disturbi agenti sulla posizione dello specchio*: sono perturbazioni della posizione dello specchio causate dalle oscillazioni del suolo (rumore sismico, rumore newtoniano), dai rumori interni alle sospensioni e allo specchio stesso (rumore termico, creep del metallo) o dovute al rumore iniettato dall'elettronica dei controlli attivi.
- *rumore di lettura*: sono disturbi intrinseci associati al sistema di lettura dell'interferometro (lo shot-noise del laser, variazioni nella pressione di radiazione, fluttuazioni in frequenza o in potenza del laser).

La somma dei vari rumori produce sui fotodiodi di lettura uno sfasamento $\delta\varphi(t)$ indistinguibile da quello prodotto da un'onda gravitazionale di am-

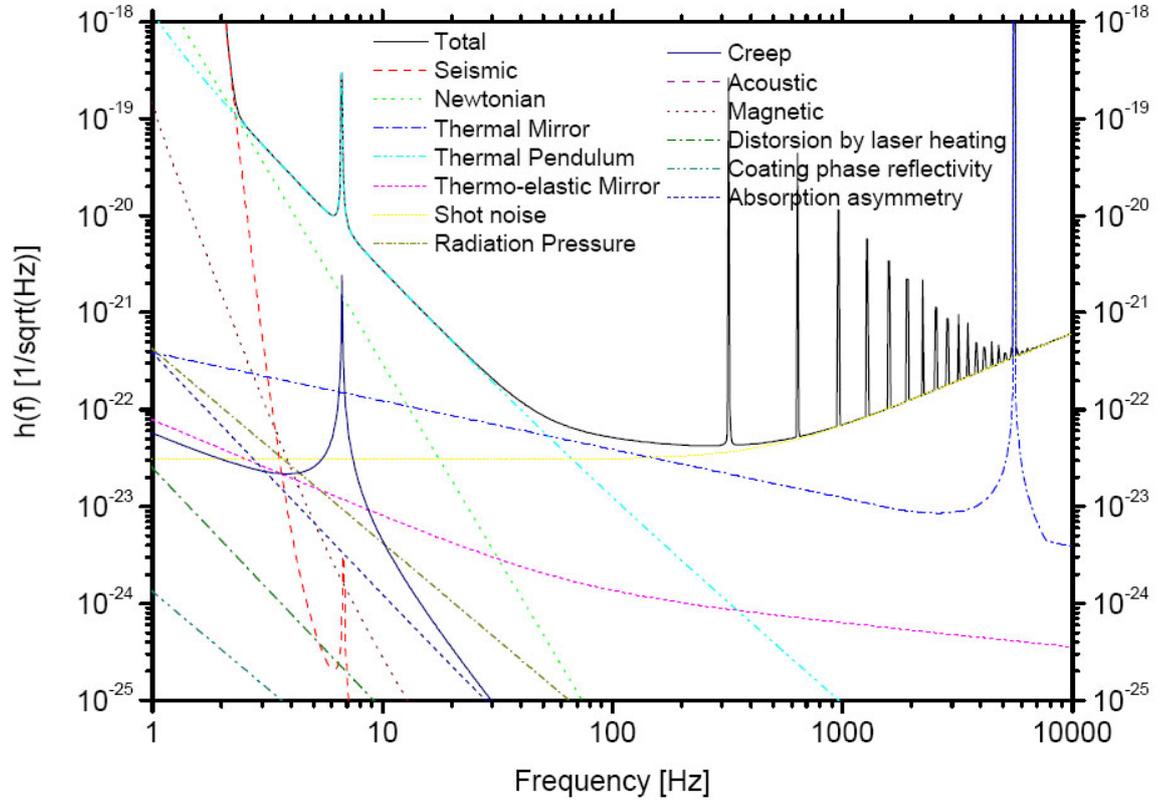


Figura 2.10: Curva di sensibilità di Virgo

piezza h_n . E' utile valutare i vari contributi del rumore nel dominio della frequenza. A tal fine si definisce la funzione di densità spettrale di potenza "bilatera":

$$P_n(f) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} h_n(t)h_n(t + \tau)e^{-i2\pi f\tau} d\tau \quad (2.13)$$

Essa rappresenta la trasformata di Fourier della funzione di autocorrelazione di $h_n(t)$. In ambito sperimentale si preferisce usare la versione "unilatera", limitata alle sole frequenze positive, definita da:

$$h_n(f) \equiv \begin{cases} 2P_n(f) & \text{se } f \geq 0 \\ 0 & \text{altrove} \end{cases} \quad (2.14)$$

La somma dei rumori convertiti nelle equivalenti densità spettrali, fornisce la curva di sensibilità di Virgo. L'identificazione di un segnale prodotto da un'onda gravitazionale è ottenuta attraverso la correlazione dell'uscita dell'interferometro con una serie di forme d'onda teoriche dette *templates*. La rivelazione di un evento si traduce quindi in un problema di rapporto

segnale-rumore. Usando la curva di sensibilità, è possibile stimare i rapporti segnale-rumore associati ai vari tipi di sorgenti astrofisiche ed ottenere così l'effettivo rate di eventi osservabili in Virgo.

Capitolo 3

Controlli digitali nelle sospensioni di Virgo

In questo capitolo vengono presentati alcuni principi della teoria dei controlli con particolare attenzione al passaggio dallo spazio di Laplace allo spazio z . Vengono inoltre trattati i problemi legati alla precisione numerica nell'implementazione dei filtri digitali e le relative soluzioni adottate in Virgo. Viene infine descritta l'elettronica del controllo locale e le sue attuali prestazioni.

3.1 Controlli in retroazione

In tutti i casi in cui si fanno variare in modo automatico le grandezze manipolabili di un sistema affinché subisca l'evoluzione temporale desiderata, operiamo un controllo automatico sul sistema [10].

In ingegneria dei controlli il termine sistema indica un processo costituito da più parti interconnesse caratterizzate da variabili dipendenti dal tempo. Esse sono dette controllabili se è possibile imporvi un arbitrario andamento temporale, non controllabili o disturbi se legate a caratteristiche intrinseche del sistema. Le funzioni descritte da tali variabili sono dette segnali. La variabile indipendente della rappresentazione matematica di un segnale può essere continua o discreta.

I segnali sono detti a tempo continuo o a tempo discreto a seconda che siano rappresentati da funzioni di variabile continua o discreta. Anche il valore del segnale può essere continuo o discreto. Si distinguono quindi segnali *analogici*, quelli per cui sia il tempo che l'ampiezza sono continui, e segnali *digitali*, quelli per cui sia il tempo che l'ampiezza sono variabili discrete. Si distinguono inoltre tre grandi classi di sistemi:

- **sistemi LTI:** Sono i sistemi lineari invarianti nel tempo. Un processo

è lineare quando soddisfa il principio di sovrapposizione ossia dati

$$\begin{aligned} u'_1(t), \dots, u'_l(t) \\ u''_1(t), \dots, u''_l(t) \end{aligned} \quad (3.1)$$

due insiemi di segnali di ingresso e

$$\begin{aligned} y'_1(t), \dots, y'_h(t) \\ y''_1(t), \dots, y''_h(t) \end{aligned} \quad (3.2)$$

le uscite relative, per ogni coppia di costanti reali k_1, k_2 , ogni segnale di ingresso della forma

$$k_1 x'_1(t) + k_2 x''_1(t), \dots, k_1 x'_n(t) + k_2 x''_n(t) \quad (3.3)$$

produce un'uscita

$$k_1 y'_1(t) + k_2 y''_1(t), \dots, k_1 y'_m(t) + k_2 y''_m(t) \quad (3.4)$$

I sistemi lineari hanno inoltre la proprietà di essere descritti da equazioni differenziali lineari del tipo

$$\sum_{i=0}^n a_i D^i y(t) = \sum_{i=0}^m b_i D^i x(t) \quad (3.5)$$

dove $x(t)$, $y(t)$ sono rispettivamente ingresso e uscita del sistema e $D^i = d^i/dt^i$.

- **sistemi non lineari:** Il modello matematico che descrive il sistema non soddisfa il principio di sovrapposizione.
- **sistemi dipendenti dal tempo**

3.1.1 Trasformata di Laplace

Lo strumento matematico principale per lo studio dei sistemi lineari a tempo continuo è la trasformata di Laplace. Essa è un operatore funzionale lineare definito come

$$L[f(t)](s) = \int_{0^-}^{\infty} f(t) e^{-st} dt \quad (3.6)$$

Le proprietà principali della trasformata di Laplace sono riassunte nella seguente tabella

Sequenza	Trasformata di Laplace
$af(t) + bg(t)$	$aF(s) + bG(s)$
$\delta(t)$	1
$u(t)$	$\frac{1}{s}$
$t^n e^{at}$	$\frac{n!}{(s-a)^{n+1}}$
$f(t) * g(t)$	$F(s)G(s)$
Teorema della traslazione temporale: $L[f(t-t_0)] = e^{-t_0 s} F(s)$	
Teorema della derivata: $L[D^i f(t)] = s^i F(s) - \sum_{j=0}^{i-1} s^j D^{i-j-1} f(t) \Big _{t=0^-}$	
Teorema dell'integrale: $L[\int_0^t f(\tau) d\tau] = \frac{1}{s} F(s)$	

Il teorema della derivata è estremamente utile in quanto consente di trasformare equazioni differenziali nel tempo in equazioni algebriche nello spazio della variabile s . Se trasformiamo ambo i membri dell'equazione (3.5) otteniamo

$$\sum_{i=0}^n a_i s^i Y(s) = \sum_{i=0}^m b_i s^i X(s) + \sum_{i=1}^n a_i \sum_{j=0}^{i-1} s^j D^{i-j-1} y(t) \Big|_{t=0^-} \quad (3.7)$$

dove $X(s)$ ed $Y(s)$ sono le trasformate di Laplace di $x(t)$, $y(t)$ e si è supposto che il sistema sia causale ($X(t) = 0$ per $t < 0$). Eguagliando termine a termine abbiamo la forma della soluzione

$$Y(s) = Y_0(s) + Y_1(s) \quad (3.8)$$

dove

$$Y_0(s) = \frac{\sum_{i=1}^n a_i \sum_{j=0}^{i-1} s^j D^{i-j-1} y(t) \Big|_{t=0^-}}{\sum_{i=0}^n a_i s^i} \quad (3.9)$$

$$Y_1(s) = \frac{\sum_{i=0}^m b_i s^i}{\sum_{i=0}^n a_i s^i} X(s) \quad (3.10)$$

Se consideriamo il sistema inizialmente in quiete $Y_0(s)$ si annulla e $Y(s) = Y_1(s)$. La funzione

$$G(s) = \frac{Y_1(s)}{X(s)} = \frac{\sum_{i=0}^m b_i s^i}{\sum_{i=0}^n a_i s^i} \quad (3.11)$$

è detta funzione di trasferimento del sistema. Essa, moltiplicata per la trasformata di Laplace del segnale in ingresso, fornisce la trasformata di Laplace dell'uscita. I polinomi a numeratore e al denominatore della (3.11)

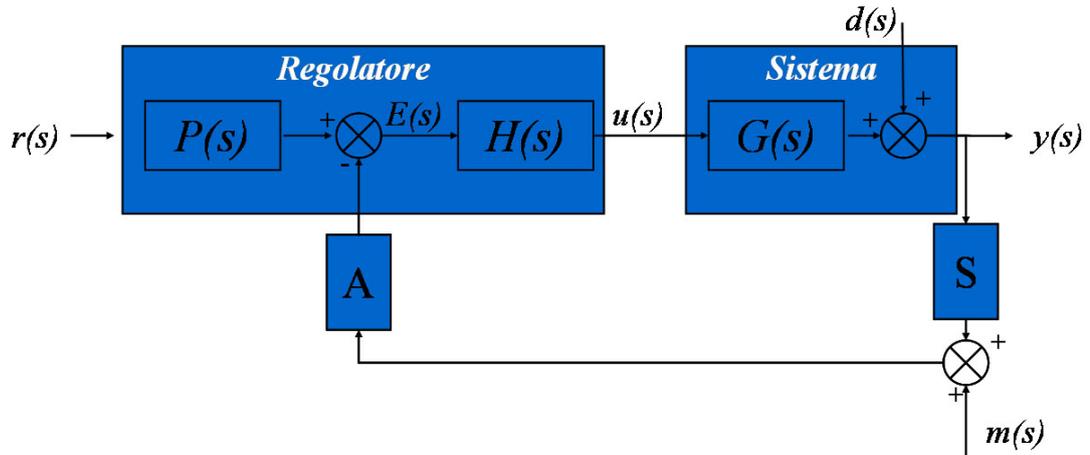


Figura 3.1: Schema tipico di un controllo in retroazione a tempo continuo

possono essere fattorizzati e quindi la funzione di trasferimento assume la forma:

$$G(s) = K \frac{(s - z_1)(s - z_2) \cdots (s - z_m)}{(s - p_1)(s - p_2) \cdots (s - p_n)} \quad (3.12)$$

dove p_1, p_2, \dots, p_n sono detti poli mentre z_1, z_2, \dots, z_m sono detti zeri della funzione di trasferimento $G(s)$. Se abbiamo una serie di sistemi in cascata è sufficiente moltiplicare le funzioni di trasferimento dei singoli blocchi per ottenere la $G(s)$ del sistema globale. E' da notare infine come nel caso di sistemi MIMO otteniamo una matrice di funzioni di trasferimento $[G(s)]$ e non una singola funzione.

3.1.2 Sistemi di controllo a tempo continuo

La fig. 3.1 mostra la configurazione tipica di un controllo in retroazione a tempo continuo. Consideriamo le parti che lo costituiscono [11]:

- $u(s)$, $y(s)$ sono rispettivamente le trasformate dei segnali d'ingresso e di uscita del sistema.
- $r(s)$ è la variabile di riferimento nello spazio di Laplace
- $E(s)$ è il segnale d'errore
- $d(s)$ indica gli ingressi non manipolabili del sistema, detti disturbi
- A ed S indicano gli attuatori ed i sensori del controllo
- $G(s)$ è la funzione o matrice di funzioni di trasferimento del sistema.

- $H(s)$ è la funzione di trasferimento del regolatore.
- $P(s)$ è la funzione di trasferimento di un generico pre-filtro. La presenza di questo termine può essere dovuta ad esempio all'elettronica di condizionamento del segnale.

L'obiettivo tipico di un sistema di regolazione è l'inseguimento, ossia l'ottenimento, nel dominio del tempo, della condizione di proporzionalità tra $y(t)$ ed $r(t)$:

$$y(t) = K \cdot r(t) \quad (3.13)$$

Tipicamente questa condizione è soddisfatta a meno di un errore

$$e(t) = K \cdot r(t) - y(t) \quad (3.14)$$

Si definisce la funzione guadagno d'anello come

$$F(s) = G(s)H(s) \quad (3.15)$$

La funzione di trasferimento del controllo ad anello chiuso risulta

$$G_c = P(s) \frac{H(s)G(s)}{1 + F(s)} \quad (3.16)$$

Un sistema è detto di tipo h a seconda della molteplicità h di un eventuale polo nell'origine di $G(s)$. Il tipo di un sistema caratterizza fortemente il comportamento asintotico per $t \rightarrow \infty$ del suo segnale d'errore $e(t)$.

Il comportamento di un buon sistema di controllo è caratterizzato da una ridotta sensibilità sia ai disturbi $d(s)$ che alla variazione dei parametri del sistema e del regolatore. Osservando la fig. 3.1, si nota come la presenza di $d(s)$ generi in uscita delle variazioni:

$$\begin{aligned} Y_d(s) &= G(s)d(s) && \text{ad anello aperto} \\ Y'_d(s) &= \frac{G(s)}{1 + G(s)H(s)}d(s) && \text{ad anello chiuso} \end{aligned} \quad (3.17)$$

Quindi i rapporti segnale-disturbo in uscita valgono nei due casi

$$\begin{aligned} \frac{Y_r(s)}{Y_d(s)} &= \frac{r(s)}{d(s)} && \text{ad anello aperto} \\ \frac{Y'_r(s)}{Y'_d(s)} &= \frac{r(s)}{d(s)}(1 + G(s)H(s)) && \text{ad anello chiuso} \end{aligned} \quad (3.18)$$

L'espressione

$$S(j\omega) = \frac{1}{1 + G(j\omega)H(j\omega)} \quad (3.19)$$

è detta funzione di sensibilità e rappresenta il fattore introdotto dalla retroazione a causa dei disturbi. Analogamente si può definire la sensibilità complementare

$$T(j\omega) = 1 - S(j\omega) = \frac{-G(j\omega)H(j\omega)}{1 + G(j\omega)H(j\omega)} \quad (3.20)$$

che esprime il rapporto tra gli errori relativi di $G(s)$ e $H(s)$ dovuti a variazioni parametriche introdotte ad esempio dall'errore di misura $m(s)$. Vale infatti

$$\frac{\Delta G_c(s)}{G_c(s)} = T(s) \frac{\Delta H(s)}{H(s)} \quad (3.21)$$

dove

$$\Delta G_c(s) = \left. \frac{\partial G_c}{\partial \alpha} \right|_{\alpha=\alpha_0} \quad \text{e} \quad \Delta H(s) = \left. \frac{\partial H}{\partial \beta} \right|_{\beta=\beta_0}$$

Combinando i vari contributi del diagramma a blocchi otteniamo la relazione ingresso-uscita:

$$Y(s) = S(s)d(s) + G_c(s)r(s) - T(s)m(s) \quad (3.22)$$

Stabilità

Requisito fondamentale di ogni sistema controllo in retroazione è la stabilità. Un sistema è detto stabile quando, inviando in ingresso una perturbazione $x(t)$ limitata in ampiezza, otteniamo un'uscita $y(t)$ limitata. Se vale la precedente condizione ed inoltre $\lim_{t \rightarrow \infty} y(t) = 0$ il sistema è detto asintoticamente stabile. Si può dimostrare, usando le proprietà della trasformata di Laplace, che condizione necessaria e sufficiente affinché un sistema sia stabile è che $G(s)$ non presenti alcun polo a parte reale positiva e che gli eventuali poli a parte reale nulla abbiano molteplicità 1. Per la stabilità asintotica è invece necessario e sufficiente che tutti i poli di $G(s)$ abbiano parte reale negativa. Rappresentando $G(s)$ nel piano complesso della variabile s , quest'ultima affermazione equivale quindi a dire che un sistema è asintoticamente stabile se e solo se tutti i poli della sua funzione di trasferimento sono contenuti nel semipiano sinistro.

Nota la stabilità del sistema di controllo ad anello aperto, sorge il problema di valutare la stabilità ad anello chiuso. A tal fine è necessario studiare il diagramma polare del guadagno d'anello $F(j\omega)$ tracciato per $-\infty < \omega < +\infty$. Vale infatti il seguente teorema noto come *criterio di Nyquist*:

- Un sistema in retroazione è asintoticamente stabile ad anello chiuso se e solo se il diagramma polare di $F(j\omega)$ tracciato per $-\infty < \omega < +\infty$ circonda il punto critico $(-1,0)$ per tanti giri al finito in senso antiorario quanti sono i poli di $F(s)$ a parte reale positiva e per tanti mezzi giri al

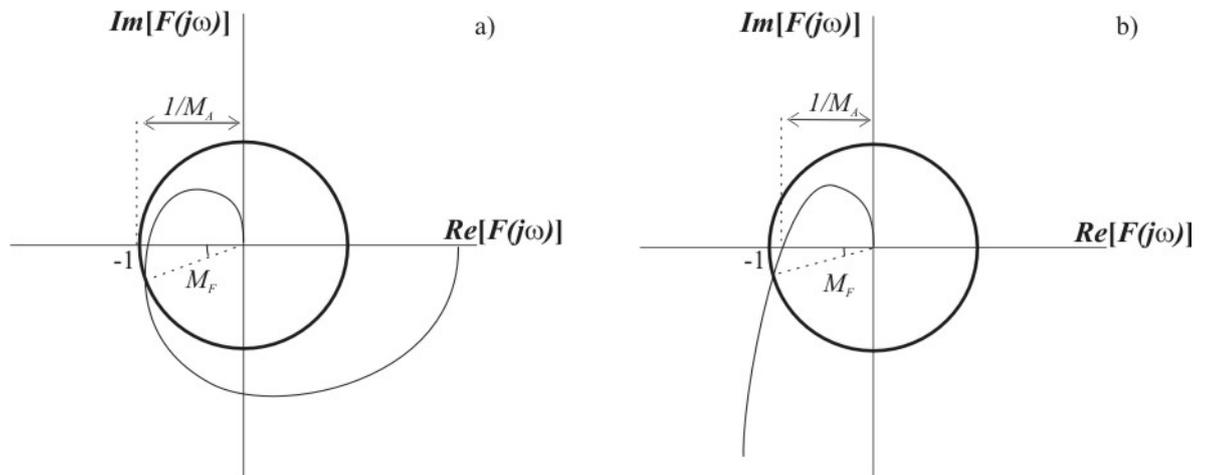


Figura 3.2: Diagramma polare del guadagno d'anello $F(j\omega)$ per un sistema di tipo 0 fig. a) e di tipo 1 fig. b)

finito in senso antiorario quanti sono i poli sull'asse immaginario. [Per giro "al finito" s'intende una rotazione del vettore $F(j\omega)$ per valori finiti non nulli del suo modulo.]

Dal diagramma polare di $F(j\omega)$, detto diagramma di Nyquist, è possibile trarre informazioni sulla criticità del sistema di controllo ossia sulla sua tendenza all'instabilità al variare dei parametri. A tal fine si definiscono due parametri detti *margini di stabilità*:

- *Margine di ampiezza:*

$$M_A = \frac{1}{|F(j\omega_\pi)|} \quad (3.23)$$

dove ω_π è la pulsazione corrispondente alla fase $-\pi$ ossia $\arctan \frac{\text{Im}[F(j\omega_\pi)]}{\text{Re}[F(j\omega_\pi)]} = -\pi$. Il margine di ampiezza viene generalmente espresso in dB.

- *Margine di fase:*

$$M_F = \arctan \frac{\text{Im}[F(j\omega_I)]}{\text{Re}[F(j\omega_I)]} + \pi \quad (3.24)$$

dove $\omega_I/2\pi$, detta frequenza di guadagno unitario f_U , è tale che $|F(j\omega_I)| = 1$.

Come mostra la figura 3.2, M_A ed M_F si determinano facilmente dal diagramma di Nyquist considerando rispettivamente l'intersezione della curva con l'asse reale e con la circonferenza di raggio unitario. Sempre dalla fig. 3.2 è possibile notare come mentre per un sistema di tipo 0 il guadagno d'anello

è rappresentato nel piano polare da una curva chiusa, per sistemi di tipo superiore presenta rami divergenti. Valori tipici per un buon comportamento di un generico sistema in retroazione sono $M_A > 6$ dB e $M_F > \frac{\pi}{4}$.

3.1.3 Trasformata Z

Nel campo dei controlli a tempo discreto il ruolo della trasformata di Laplace è ricoperto dalla trasformata Z [12]. La trasformata Z di una sequenza numerica $x(n)$ è definita come

$$X(z) = \sum_{n=-\infty}^{+\infty} x(n)z^{-n} \quad (3.25)$$

dove $z \in \mathbb{C}$. La (3.25) è una serie di Laurent e dunque $X(z)$ è una funzione analitica in ogni punto interno alla regione di convergenza. Usando il teorema di Cauchy è possibile definire la corrispondente antitrasformata

$$x(n) = \frac{1}{2\pi i} \oint_{\gamma} X(z)z^{n-1} dz \quad (3.26)$$

dove γ è un percorso chiuso che circonda l'origine del piano z ed è all'interno della regione di convergenza della serie. Le principali proprietà della trasformata Z sono mostrate nella seguente tabella:

Sequenza	Trasformata Z
$ax(n) + by(n)$	$aX(z) + bY(z)$
$x(n + n_0)$	$z^{n_0}X(z)$
$x(-n)$	$X(1/z)$
$a^n x(n)$	$X(a^{-1}z)$
$nx(n)$	$-z \frac{dX(z)}{dz}$
$x(n) * y(n)$	$X(z)Y(z)$
$\lim_{n \rightarrow \infty} x(n) = \lim_{z \rightarrow 1} [(1 - z^{-1})X(z)]$ $x(0) = \lim_{z \rightarrow \infty} X(z)$	

Una importante classe di sistemi discreti è costituita da quei sistemi LTI per i quali l'ingresso $x(n)$ e l'uscita $y(n)$ soddisfano un'equazione alle differenze della forma

$$\sum_{k=0}^N a_k y(n-k) = \sum_{r=0}^M b_r x(n-r) \quad (3.27)$$

Usando le proprietà della trasformata Z, si può vedere che la (3.27) è l'equivalente discreto dell'equazione differenziale (.). Come la funzione di trasferimento analogica $G(s)$ esprime il rapporto tra le trasformate di Laplace dei

segnali d'ingresso e d'uscita, è possibile definire, applicando la trasformata Z alla (3.27), una funzione di trasferimento digitale:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{r=0}^M b_r z^{-r}}{\sum_{k=0}^N a_k z^{-k}} \quad (3.28)$$

Fattorizzando numeratore e denominatore della precedente espressione otteniamo l'equivalente discreto della (3.12)

$$H(z) = A \frac{\prod_{r=1}^M (1 - c_r z^{-1})}{\prod_{k=1}^N (1 - d_k z^{-1})} \quad (3.29)$$

Ciascuno dei fattori del numeratore contribuisce con un zero in $z = c_r$ ed un polo $z = 0$ mentre ogni fattore del denominatore dà luogo ad un polo in $z = d_k$ e ad uno zero in $z = 0$.

Attraverso la trasformata Z, molte delle proprietà dei sistemi a tempo continuo possono essere estese ai sistemi a tempo discreto. Ad esempio consideriamo la sequenza

$$x(n) = a^n u(n)$$

dove $u(n)$ è il gradino unitario e deve essere $a < 1$ affinché la sequenza abbia ampiezza finita. Utilizzando le proprietà riportate nella tabella, si può vedere che la trasformata Z di $x(n)$ vale

$$X(z) = \sum_0^{\infty} (az^{-1})^n = \frac{1}{1 - az^{-1}} \quad \text{per } |z| > |a|$$

Il denominatore dell'ultimo termine della precedente espressione corrisponde ad uno dei fattori $(1 - d_k z^{-1})$ della (3.29). Quindi a altro non è che un polo di una generica funzione di trasferimento $H(z)$. L'ampiezza finita della sequenza $x(n)$, che vale solo se il polo $a < 1$, è quindi condizione necessaria affinché il sistema descritto da $H(z)$ sia stabile. Rappresentando la funzione di trasferimento nel piano complesso della variabile z , detto piano z , questa affermazione equivale a dire che i poli di un sistema stabile devono essere contenuti in un cerchio di raggio unitario.

3.1.4 Sistemi di controllo a tempo discreto

Nei sistemi di controllo digitali viene usato un regolatore della forma mostrata in fig. 3.3. Il segnale d'errore $E(s)$, dopo essere stato convertito in una sequenza numerica da un ADC, viene filtrato da un calcolatore e riconvertito in forma analogica da un DAC. La presenza di un calcolatore programmabile rappresenta il principale vantaggio dei sistemi di controllo digitale rispetto a quelli analogici. Infatti, modificando il codice eseguito dall'elaboratore, è possibile cambiare completamente le caratteristiche del

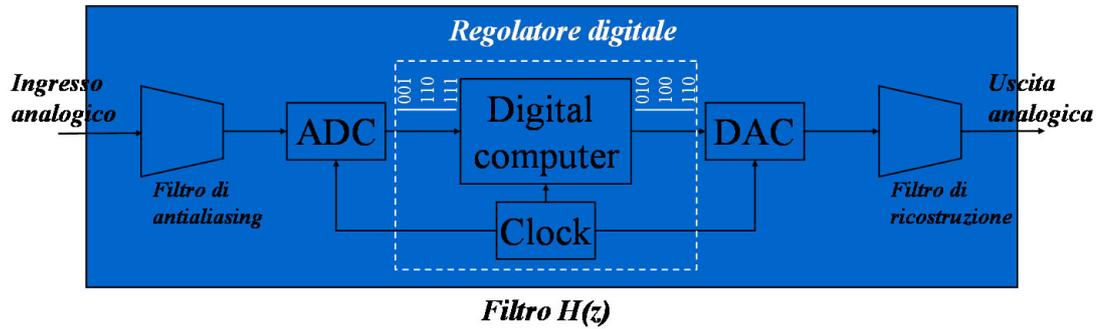


Figura 3.3: Schema tipico di un regolatore digitale

sistema senza doverne alterare l'elettronica. Come vedremo successivamente, infatti, usando la (3.27), è possibile implementare un filtro generico di ordine N sotto forma di algoritmo numerico.

Il processo di conversione analogico-digitale pone tuttavia alcuni problemi. Nei sistemi digitali il segnale d'errore $E(s)$ deve essere campionato e quantizzato. Il processo di campionamento o *sampling* trasforma il segnale continuo in ingresso in un segnale a tempo discreto. Questa funzione è svolta da un circuito detto *Sample and Hold*, integrato nella maggioranza degli ADC, che riceve in ingresso un segnale analogico e lo mantiene ad un valore costante per un tempo T_S detto periodo di campionamento. La distorsione del segnale introdotta dal campionamento si traduce in un'ambiguità nella rappresentazione delle componenti ad alta frequenza del segnale stesso. Risulta infatti impossibile ricostruire univocamente, partendo dalla sua versione campionata, un segnale contenente componenti con frequenza maggiore di $f_c/2$, dove $f_c = 1/T_S$. Questa affermazione è detta teorema del campionamento ed è stata formulata da C.Shannon e H.Nyquist nel 1949. Nella realizzazione di un sistema digitale è quindi necessario l'uso di filtri passa-basso all'ingresso dell'ADC (filtri di antialiasing) e all'uscita del DAC (filtri di ricostruzione) con frequenza di taglio minore di $f_c/2$.

Nei sistemi digitali al processo di campionamento segue la quantizzazione ossia la conversione in numeri binari dell'ampiezza del segnale [13]. Non tutti i bit in una stringa binaria sono ugualmente significativi: si definisce infatti MSB (Most Significant Bit) il bit più a sinistra nella stringa ed LSB (Least Significant Bit) il bit più a destra. Quest'ultimo è detto anche livello di quantizzazione Q e fornisce una misura dell'errore di troncamento E_T che si commette nel rappresentare l'ampiezza con un numero finito di bit. Vale infatti

$$0 \leq |E_T| \leq \frac{1}{2}Q \quad (3.30)$$

L'effetto dell'errore E_T è quello di introdurre rumore, detto rumore di quan-

tizzazione, che si somma a quello tipicamente già presente nel segnale analogico. Supponendo che la funzione di densità di probabilità $f(\xi)$ sia uniformemente distribuita tra $-\frac{1}{2}Q$ e $+\frac{1}{2}Q$, possiamo calcolare la varianza del rumore di quantizzazione:

$$\sigma^2 = \mathcal{E}[E_T - \overline{E_T}]^2 = \int_{-\infty}^{+\infty} \xi^2 \frac{1}{Q} d\xi = \frac{Q^2}{12} \quad (3.31)$$

3.2 Filtri digitali in Virgo

A seconda del valore di N nella (3.27), si distinguono due importanti classi di filtri: filtri con risposta all'impulso finita nel tempo (*FIR*), se $N = 0$, e filtri con risposta all'impulso infinita nel tempo (*IIR*), se $N > 0$. Ogni sistema LTI a tempo discreto può essere infatti descritto da un'equazione del tipo

$$y(n) = \sum_{k=-\infty}^{+\infty} T[\delta(n)]x(n-k) = T[\delta(n)] * x(n) \quad (3.32)$$

dove T è l'operatore lineare che trasforma l'ingresso in uscita, $y(n) = T[x(n)]$, e $\delta(n)$ è l'impulso unitario. Quindi analogamente ai sistemi a tempo continuo, in cui l'uscita può essere espressa come integrale di convoluzione tra l'ingresso e la funzione di Green del sistema, nei sistemi discreti l'uscita è pari alla somma di convoluzione tra l'ingresso e la risposta all'impulso $T[\delta(n)]$. Da ciò si deduce ad esempio che un sistema LTI, per essere stabile, deve verificare la condizione necessaria e sufficiente:

$$S = \sum_{k=-\infty}^{+\infty} T[\delta(n)] < \infty \quad (3.33)$$

I filtri IIR sono detti anche ricorsivi in quanto alcuni valori precedenti dell'uscita insieme ai valori passati e presenti dell'ingresso determinano l'uscita attuale del filtro. Nei filtri FIR invece, l'uscita dipende solo dai valori presenti e passati dell'ingresso. In Virgo vengono implementati esclusivamente filtri IIR.

3.2.1 Trasformazione bilineare

Uno dei metodi principali usati nella progettazione dei filtri digitali consiste nel calcolare la funzione di trasferimento del filtro nello spazio di Laplace, considerando il sistema a tempo continuo, e nell'operare una particolare trasformazione dallo spazio s allo spazio z . La tecnica matematica utilizzata nel controllo delle sospensioni di Virgo, che consente tale passaggio, è la trasformazione bilineare.

Consideriamo un sistema descritto da un'equazione differenziale del primo ordine:

$$c_1 y'_a(t) + c_a y_a(t) = d_0 x(t) \quad (3.34)$$

Facendo la trasformata di Laplace di ambo i membri otteniamo la funzione di trasferimento analogica del sistema:

$$H_a(s) = \frac{d_0}{c_1 s + c_0} \quad (3.35)$$

Poiché chiaramente

$$y_a(t) = \int_{t_0}^t y'_a(t) + y_a(t_0) \quad (3.36)$$

ponendo $t = nT$ e $t_0 = (n-1)T$, dove $1/T$ è la frequenza di campionamento, ed approssimando l'integrale (3.36) col metodo dei trapezi, otteniamo

$$y_a(nT) = y_a((n-1)T) + \frac{T}{2} [y'_a(nT) + y'_a((n-1)T)] \quad (3.37)$$

Combinando questa espressione con la (3.34), abbiamo

$$[y(n) - y(n-1)] = \frac{T}{2} \left[-\frac{c_0}{c_1} (y(n) + y(n-1)) + \frac{d_0}{c_1} (x(n) + x(n-1)) \right] \quad (3.38)$$

dove $y(n) = y_a(nT)$ e $x(n) = x_a(nT)$. Facendo adesso la trasformata Z e ricavando $H(z)$ si ha

$$H(z) = \frac{Y(z)}{X(z)} = \frac{d_0}{c_1 \frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}} + c_0} \quad (3.39)$$

E' quindi possibile ricavare $H(z)$ da $H(s)$ attraverso la sostituzione:

$$s = \frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}} \quad (3.40)$$

La (3.40) è detta trasformazione bilineare. Questo risultato ha validità molto generale in quanto ogni equazione differenziale lineare di ordine N può essere espressa come sistema di N equazioni del primo ordine della forma (3.34).

Da un punto di vista analitico la trasformazione bilineare è una trasformazione conforme che mappa il piano s nel piano z (fig. 3.4): il semipiano sinistro è mappato all'interno del cerchio di raggio unitario, il semipiano destro all'esterno del cerchio e l'asse immaginario sulla circonferenza. Per verificare quest'ultima affermazione poniamo $z = e^{j\omega}$ nella (3.40) ottenendo

$$s = a + j\omega = \frac{2}{T} i \tan(\omega/2) \Rightarrow a = 0 \quad (3.41)$$

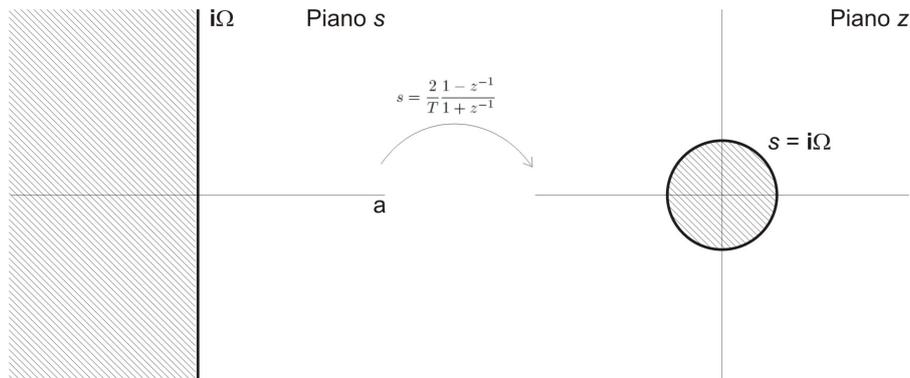


Figura 3.4: Mapping del piano s nel piano z tramite la trasformazione bilineare

Invertendo la precedente espressione abbiamo

$$\omega = 2 \tan^{-1} \left(\frac{\Omega T}{2} \right) \quad (3.42)$$

da cui risulta che il semiasse immaginario superiore è mappato nella metà superiore della circonferenza unitaria, mentre la parte negativa nella metà inferiore.

Una conseguenza importante della (3.42) è che le frequenze, e quindi le posizioni dei poli e degli zeri del filtro analogico, vengono distorte dalla trasformazione bilineare. Tuttavia il requisito fondamentale che filtri analogici stabili vengano convertiti in filtri digitali stabili, viene rispettato. Infatti, come abbiamo detto nel par. 3.1.2, i poli di un sistema stabile a tempo continuo devono essere contenuti nel semipiano sinistro dello spazio s . Analogamente, come abbiamo visto nel par. 3.1.3, in un sistema stabile a tempo discreto i poli devono essere contenuti nella circonferenza di raggio unitario del piano z . La trasformazione bilineare mappa queste due regioni una nell'altra conservando quindi la stabilità dei filtri.

Per ovviare alla distorsione delle frequenze è sufficiente usare la (3.42) per predistorgere la funzione di trasferimento del filtro analogico, *prewarping* in gergo, in modo da ottenere il filtro digitale desiderato.

3.2.2 Precisione numerica

Ottenuta la funzione di trasferimento $H(z)$ del filtro digitale è possibile tradurla in un algoritmo numerico da implementare in un calcolatore riscrivendo la (3.28) come

$$H(z) = A \frac{1 + \sum_{k=1}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}} \quad (3.43)$$

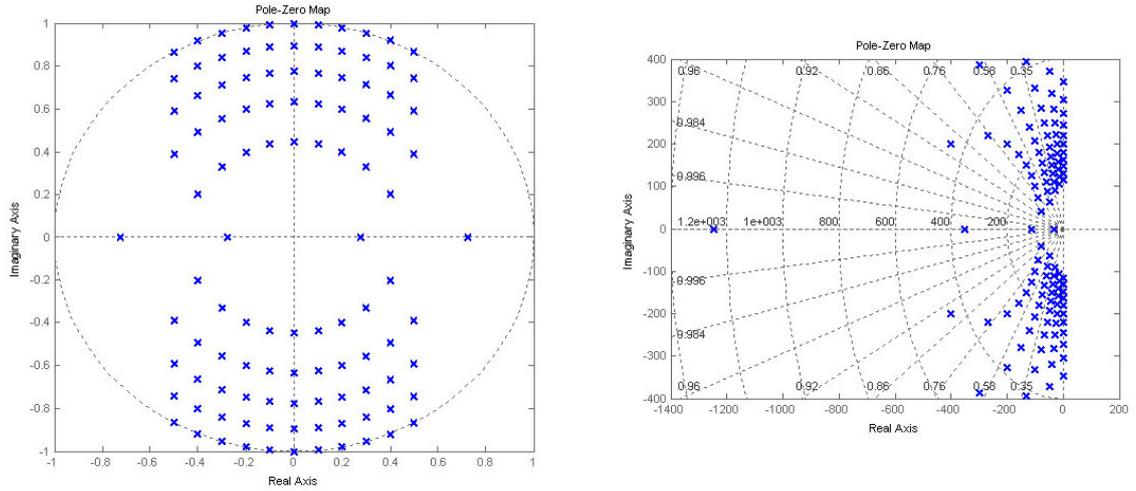


Figura 3.5: Risposta in frequenza di un filtro del II ordine con coefficienti quantizzati a 5 bit.

dove A è un coefficiente di guadagno del filtro. Ponendo successivamente

$$\frac{Y(z)}{X(z)} = A \frac{Y(z)}{W(z)} \frac{W(z)}{X(z)} \quad (3.44)$$

$$\frac{W(z)}{X(z)} = 1 / \left(1 + \sum_{k=1}^N a_k z^{-k} \right) \quad (3.45)$$

$$\frac{Y(z)}{W(z)} = A \left(1 + \sum_{k=1}^M b_k z^{-k} \right) \quad (3.46)$$

Dalle precedenti equazioni si ricava un equivalente sistema di equazioni alle differenze:

$$\begin{cases} w_n = x_n - \sum_{k=1}^N a_k w_{n-k} \\ y_n = A \left(w_n + \sum_{k=1}^M b_k w_{n-k} \right) \end{cases} \quad (3.47)$$

A partire dall'ingresso x_n e dallo stato interno del filtro, costituito dagli N w_k , si possono quindi calcolare lo stato interno e l'uscita del filtro al ciclo successivo con $N + M + 1$ prodotti e $N + M$ somme [14].

Una volta ottenuto l'algoritmo sorge il problema della precisione numerica. Infatti la funzione di trasferimento del filtro digitale viene ricavata assumendo una precisione illimitata nella rappresentazione dei coefficienti. Tuttavia qualsiasi calcolatore su cui verrà implementato l'algoritmo ha una lunghezza di parola finita. L'errore di quantizzazione introdotto si traduce in una deviazione dalla risposta in frequenza desiderata del filtro e può avere quindi conseguenze catastrofiche sul sistema di controllo progettato:

ad esempio, nel caso dei filtri IIR, uno o più poli possono spostarsi al di fuori del cerchio unitario del piano z rendendo il sistema instabile. Un esempio di questi effetti è mostrato in fig. 3.5: un filtro del II ordine con $H(z) = \frac{1}{1 - 2r \cos \theta z^{-1} + r^2 z^{-2}}$ possiede due poli per $z = re^{\pm i\theta}$. Tuttavia, quantizzando i coefficienti a 5 bit, possiamo implementare solo poli in un numero finito di posizioni: quelle rappresentate dalle \times nella figura. Il problema è particolarmente rilevante nel caso di Virgo in quanto, come vedremo, al fine di smorzare i modi normali del SA, si usano filtri con poli e zeri a bassa frequenza. Come si può vedere dalla (3.42), la regione del piano s corrispondente a piccoli valori di ω viene mappata dalla trasformazione bilineare in una regione di ridotta ampiezza angolare attorno all'asse reale del piano z . Come notiamo dalla fig. 3.5, questa è proprio la regione in cui la griglia di possibili posizioni dei poli è meno densa.

Data la loro importanza, è quindi utile analizzare le varie forme di rappresentazione numerica usate nei calcolatori.

3.2.3 Rappresentazione numerica

Sebbene vi siano molti possibili metodi per la codifica dei numeri in un calcolatore, due tipi di rappresentazioni vengono essenzialmente usati:

Rappresentazione in virgola fissa (*fixed point*) : è usata in particolare per rappresentare i numeri interi. Un semplice cambiamento di base, da decimale a binario, non consentirebbe di codificare gli interi negativi. Per risolvere il problema tre formati alternativi vengono comunemente usati:

- **Sign and magnitude**: Data una stringa $\{x_{N-1}, x_{N-2}, \dots, x_0\}$, viene scelto il bit più significativo x_{N-1} per codificare il segno ed usati i bit restanti per il valore binario del modulo del numero: dato un intero X , il valore aritmetico corrispondente vale

$$X = (1 - 2x_{N-1}) \sum_{i=0}^{N-2} x_i 2^i \quad (3.48)$$

I problemi principali sono che una configurazione di bit viene sprecaata, datoché vi sono due rappresentazioni dello zero ($+0 = 0000$ e $-0 = 1000$), e che l'aritmetica dei numeri in questo formato si implementa male.

- **Offset binary**: In questo schema usato in alcuni ADC e DAC, viene sottratto ad ogni numero una costante, detta *offset*, pari alla metà del massimo numero rappresentabile:

$$X = \sum_{i=0}^{N-1} x_i 2^i - \sum_{i=0}^{N-2} 2^i \quad (3.49)$$

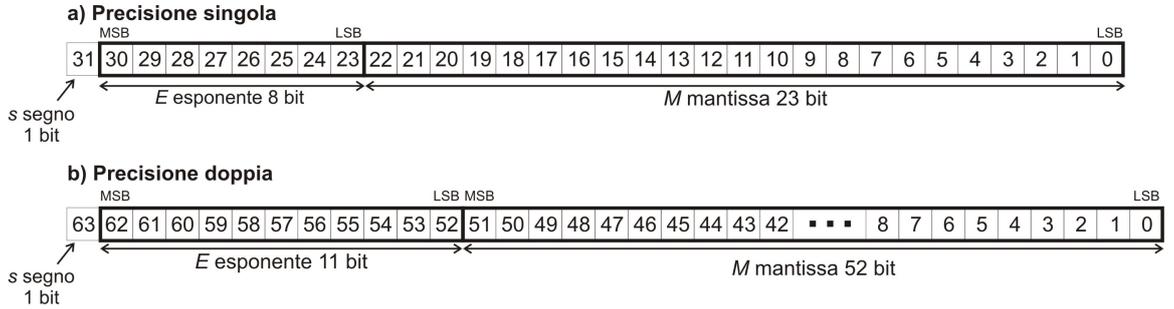


Figura 3.6: Formati in virgola mobile a precisione singola e doppia

Non essendo un formato standardizzato possono essere definiti anche altri possibili offset.

- **Complemento a 2:** è il formato usato nei processori dei PC e in molti convertitori ADC e DAC tra cui quelli attualmente usati in Virgo (v. cap.). Dato l'intero X , il valore rappresentato della stringa di bit vale:

$$X = \sum_{i=0}^{N-2} x_i 2^i - x_{N-1} 2^{N-1} \quad (3.50)$$

Il bit più significativo X_{N-1} indica il segno del numero (0 positivo, 1 negativo). I numeri positivi sono codificati con un semplice cambiamento di base, i numeri negativi, convertendo in binario il valore assoluto dell'intero, facendo il complemento bit per bit (complemento a 1), ed aggiungendo 1 (complemento a 2). E' quindi chiaro che un numero in complemento a due può essere ottenuto da un offset binary facendo il complemento del bit più significativo. La praticità di questo formato risiede nella semplicità con cui si implementano le operazioni aritmetiche.

Rappresentazione in virgola mobile (*floating point*) : essa è essenzialmente un trasposizione in base 2 della notazione scientifica usata per i numeri reali. Sebbene anche in questo caso siano possibili più formati, lo standard ANSI/IEEE 754-1985 è il riferimento più comunemente adottato. Esso definisce un formato numerico a 32 bit detto precisione singola ed un formato a 64 bit detto precisione doppia. Vi sono poi due formati "ibridi" detti precisione singola estesa e precisione doppia estesa che possono essere implementati in modo diverso a seconda dell'applicazione. La fig. 3.6 a) mostra la suddivisione della stringa in 3 gruppi di bit prevista per la precisione singola: il bit 31 definisce il segno del numero, 8 bit, dal 30 al 23, rappresentano

l'esponente e i restanti 23 bit la mantissa [15]. L'equivalente raggruppamento usato per la precisione doppia è mostrato in fig. 3.6 b). In questo modo il numero desiderato è ottenuto attraverso la relazione:

$$\nu = (-1)^s \cdot M \cdot 2^{E-bias} \quad (3.51)$$

Dall'esponente E viene sottratta una costante, detta *bias*, che vale 127 per il formato singolo, 1023 per il doppio, al fine di introdurre gli esponenti negativi. Il numero M indica la mantissa e vale

$$M = \sum_{i=0}^{p-1} b_i 2^{-i} = b_0 \cdot b_1 \cdots b_{p-1} \quad (3.52)$$

con $b_i = 0$ o $1 \forall i \neq 0$. Si assume $b_0 = 1$ in modo da avere, come nella notazione scientifica decimale, una sola cifra diversa da zero a sinistra della virgola. I valori dei parametri per i vari formati sono riassunti nella seguente tabella

Parametro	Formato			
	Singolo	Singolo esteso	Doppio	Doppio esteso
p	24	≥ 32	53	≥ 64
E_{max}	+127	$\geq +1023$	+1023	$\geq +16383$
E_{min}	-126	≤ -1022	-1022	≤ -16382
<i>bias</i>	+127	non definito	+1023	non definito
Larghezza dell'esponente in bit	8	≥ 11	11	≥ 15
Larghezza del formato in bit	32	≥ 43	64	≥ 79

Valgono inoltre le seguenti convenzioni:

- 1) Se tutti i bit dell'esponente sono 1, ossia $E = 255$ o $E = 2047$ a seconda del formato, ed $M = 0$, allora $\nu = (-1)^s \infty$
- 2) Se $E = 0$ ed $M = 0$ allora $\nu = (-1)^s 0$
- 3) Se tutti i bit dell'esponente sono 1 e $M \neq 0$ la stringa di bit è detta *NaN* (Not A Number) in quanto non le viene assegnato alcun significato.
- 4) Se $E = 0$ ed $M \neq 0$ allora $\nu = (-1)^s \cdot 2^{-(bias-1)} \cdot (0. M)$ Questi numeri a precisione minore sono detti non normalizzati poiché ottenuti eliminando la condizione $b_0 = 1$.

In questo modo il numero massimo e minimo rappresentabili valgono rispettivamente $\pm 3.4 \cdot 10^{38}$ e $\pm 1.2 \cdot 10^{-38}$ in precisione singola e addirittura $\pm 1.8 \cdot 10^{308}$ e $\pm 2.2 \cdot 10^{-308}$ in precisione doppia. Infine la precisione relativa con cui si può approssimare un numero è data da

$$\epsilon = 2^{1-p} \quad (3.53)$$

e vale quindi

$$\epsilon = \begin{cases} 2^{-23} \simeq 1.19 \cdot 10^{-7} & \text{in precisione singola} \\ 2^{-52} \simeq 2.22 \cdot 10^{-16} & \text{in precisione doppia} \end{cases} \quad (3.54)$$

3.2.4 Forme di implementazione

Un filtro digitale può essere rappresentato come rete di elaborazione numerica in due forme alternative:

- **forma diretta:** come già operato in precedenza, si passa da $H(z)$ al corrispondente sistema di equazioni alle differenze e si implementa l'algoritmo
- **forma in cascata:** la funzione di trasferimento razionale $H(z)$ è scomposta in fattori e riscritta come

$$H(z) = \frac{\prod_{k=1}^{M_1} (1 - g_k z^{-1}) \prod_{k=1}^{M_2} (1 - h_k z^{-1})(1 - h_k^* z^{-1})}{\prod_{k=1}^{N_1} (1 - c_k z^{-1}) \prod_{k=1}^{N_2} (1 - d_k z^{-1})(1 - d_k^* z^{-1})} \quad (3.55)$$

con $M = M_1 + 2M_2$ ed $N = N_1 + 2N_2$. Vengono quindi realizzati una serie di filtri in cascata del I ordine con zeri reali g_k e poli reali c_k ed una serie del II ordine con zeri complessi coniugati h_k e h_k^* e poli d_k , d_k^* .

Sebbene entrambe le forme rappresentino la stessa funzione di trasferimento $H(z)$, ciascuna di esse ha una diversa sensibilità alla quantizzazione dei parametri. Ritornando all'equazione e riscrivendola come

$$\widehat{H}(z) = A \frac{\sum_{k=0}^M \widehat{b}_k z^{-k}}{1 - \sum_{k=1}^N \widehat{a}_k z^{-k}} \quad (3.56)$$

dove $\widehat{a}_k = a_k + \Delta a_k$ e $\widehat{b}_k = b_k + \Delta b_k$ sono i coefficienti approssimati a causa della precisione numerica. Supponendo i poli della funzione di trasferimento nelle posizioni $z = z_i$ con $i = 1, 2, \dots, N$, è possibile esprimere il denominatore di $H(z)$ nella forma

$$P(z) = 1 - \sum_{k=1}^N a_k z^{-k} = \prod_{k=1}^N (1 - z_k z^{-1}) \quad (3.57)$$

Definendo i poli di $\widehat{H}(z)$ come $z_i + \Delta z_i$ con $i = 1, 2, \dots, N$, è possibile scrivere, usando la teoria della propagazione degli errori

$$\Delta z_i = \sum_{k=1}^N \frac{\partial z_i}{\partial a_k} \Delta a_k \quad (3.58)$$

Datoché inoltre risulta

$$\left(\frac{\partial P(z)}{\partial z_i} \right)_{z=z_i} \frac{\partial z_i}{\partial a_k} = \left(\frac{\partial P(z)}{\partial a_k} \right)_{z=z_i} \quad (3.59)$$

segue che

$$\frac{\partial z_i}{\partial a_k} = \frac{z_i^{N-k}}{\prod_{\substack{l=1 \\ l \neq i}}^N (z_i - z_l)} \quad (3.60)$$

Osservando la (3.60) è subito evidente come al crescere dell'ordine N del filtro peggiori il suo comportamento rispetto alla quantizzazione dei coefficienti. Usando la (3.60) è possibile inoltre dimostrare che se i poli sono fittamente raggruppati può accadere che piccoli errori nei coefficienti creino grandi spostamenti nei poli. Infatti, osservando il denominatore della (3.60), notiamo come ogni fattore $(z_i - z_l)$ della produttoria possa essere espresso come vettore nel piano Z . Quindi il modulo del denominatore risulta uguale al prodotto delle distanze degli z_l rispetto al polo i -esimo.

Da queste considerazioni segue che una configurazione in cascata, realizzando ciascuna coppia di poli complessi coniugati separatamente, ha, rispetto alla forma diretta del filtro, una sensibilità minore alla quantizzazione dei parametri. In Virgo vengono implementati solo filtri in cascata del primo e del secondo ordine.

3.2.5 Filtri del I ordine

Consideriamo la funzione di trasferimento analogica di un filtro del primo ordine:

$$H(s) = A \frac{s + \omega_z}{s + \omega_p} \quad (3.61)$$

dove ω_z e ω_p sono rispettivamente le frequenze angolari dello zero e del polo del filtro. Usando la trasformazione bilineare otteniamo

$$H(z) = A \frac{2 + T\omega_z}{2 + T\omega_p} \frac{1 - \frac{2-T\omega_z}{2+T\omega_z} z^{-1}}{1 - \frac{2-T\omega_p}{2+T\omega_p} z^{-1}} = B \frac{1 + az^{-1}}{1 + bz^{-1}} \quad (3.62)$$

Se per rappresentare i coefficienti a e b nella memoria di un calcolatore usiamo numeri a virgola mobile, la loro precisione relativa è data dalla ϵ definita in (3.53). Poiché la frequenza di campionamento $1/T$ è molto minore delle frequenze $\omega_z/2\pi$ e $\omega_p/2\pi$, risulta che $a, b \simeq -1$ e che quindi i coefficienti sono rappresentati con precisione assoluta ϵ . L'equivalente della (3.58) nel piano s vale

$$\Delta\omega_i = \sum_{k=1}^N \left| \frac{\partial\omega_i}{\partial a_k} \right| \Delta a_k \quad (3.63)$$

Datoché come risulta dalla

$$\omega_z = \frac{2}{T} \frac{1+a}{1-a} \quad (3.64)$$

otteniamo, valutando la derivata parziale in $a = -1$,

$$\Delta f = \frac{\Delta a}{2\pi T} = \frac{\epsilon}{2\pi} f_c \quad (3.65)$$

dove f_c è la frequenza di campionamento. Poiché, come già detto, nel controllo locale di Virgo $f_c = 10$ kHz, abbiamo $\Delta f \simeq 190$ μ Hz in precisione singola e $\Delta f \simeq 0.7$ μ Hz in precisione singola estesa (la sua implementazione più comune prevede 31 bit di mantissa). Data l'esigenza di smorzare anche le componenti a bassissima frequenza del moto del superattenuatore, è stato scelta la precisione singola estesa.

3.2.6 Filtri del II ordine

La funzione di trasferimento generica di un filtro del secondo ordine vale nel piano s

$$H(s) = A \frac{s^2 + \frac{\omega_z s}{Q_z} + \omega_z^2}{s^2 + \frac{\omega_p s}{Q_p} + \omega_p^2} \quad (3.66)$$

dove ω_z e ω_p sono, come prima, le frequenze angolari degli zeri e dei poli e Q_z e Q_p i rispettivi fattori di qualità. Consideriamo a scelta solo il numeratore o il denominatore della (3.66):

$$P(s) = s^2 + \frac{\omega s}{Q} + \omega^2 \quad (3.67)$$

Facendo la trasformazione bilineare abbiamo

$$P(z) = B \frac{(1 + az^{-1} + bz^{-2})}{(1 + z^{-1})^2} \quad (3.68)$$

dove

$$\begin{cases} a = \frac{2(\omega^2 T^2 - 4)}{4 + \frac{2\omega T}{Q} + \omega^2 T^2} \\ b = \frac{4 - \frac{2\omega T}{Q} + \omega^2 T^2}{4 + \frac{2\omega T}{Q} + \omega^2 T^2} \end{cases} \quad (3.69)$$

Poiché come prima $\omega T \ll 1$, $a \simeq -2$ e $b \simeq 1$. Invertendo il sistema (3.69) si ottiene

$$\begin{cases} \omega = \frac{2}{T} \sqrt{\frac{1+b+a}{1+b-a}} \\ Q = \frac{\sqrt{(1+b+a)(1+b-a)}}{2(1-b)} \end{cases} \quad (3.70)$$

Applicando la (3.63) si ricava la precisione con cui è possibile approssimare ω e Q in funzione dell'errore sui coefficienti:

$$\begin{cases} \Delta\omega = \left| \frac{\partial\omega}{\partial a} \right| \Delta a + \left| \frac{\partial\omega}{\partial b} \right| \Delta b \\ \Delta Q = \left| \frac{\partial Q}{\partial a} \right| \Delta a + \left| \frac{\partial Q}{\partial b} \right| \Delta b \end{cases} \quad (3.71)$$

Valutando le derivate parziali in $a = -2$ e $b = 1$ otteniamo

$$\begin{cases} \left(\frac{\partial\omega}{\partial a} \right)_{a=-2, b=1} = \frac{1}{2\omega T^2}; & \left(\frac{\partial\omega}{\partial b} \right)_{a=-2, b=1} = \frac{1}{2\omega T^2} \\ \left(\frac{\partial Q}{\partial a} \right)_{a=-2, b=1} = \frac{Q}{2\omega^2 T^2}; & \left(\frac{\partial Q}{\partial b} \right)_{a=-2, b=1} = \frac{Q}{2\omega^2 T^2} \end{cases} \quad (3.72)$$

Datoché $\Delta a = 2\epsilon$ e $\Delta b = \epsilon$ sostituendo nella si ha

$$\begin{cases} \Delta\omega = \frac{3\epsilon}{2\omega T^2} \\ \Delta Q = \frac{3\epsilon Q}{2\omega^2 T^2} \end{cases} \quad (3.73)$$

Quindi la precisione relativa è uguale per ω e per Q :

$$\frac{\Delta\omega}{\omega} = \frac{\Delta Q}{Q} = \frac{3\epsilon}{2\omega^2 T^2} \quad (3.74)$$

Imponiamo adesso la condizione che la frequenza del polo o dello zero sia all'interno della banda del filtro: deve essere

$$\Delta\omega < \frac{\omega}{Q} \quad (3.75)$$

che applicata alla fornisce

$$F = \frac{\omega}{2\pi} > S \frac{f_c}{2\pi} \sqrt{\frac{3\epsilon Q}{2}} \quad (3.76)$$

o in alternativa

$$Q < \frac{2\omega^2}{3\epsilon f_c S^2} \quad (3.77)$$

La (3.77) pone un limite superiore al fattore di qualità ottenibile in funzione della frequenza per una data precisione numerica. S rappresenta un generico fattore di sicurezza, stimato empiricamente, introdotto affinché il rapporto segnale-rumore del filtro sia sufficientemente elevato.

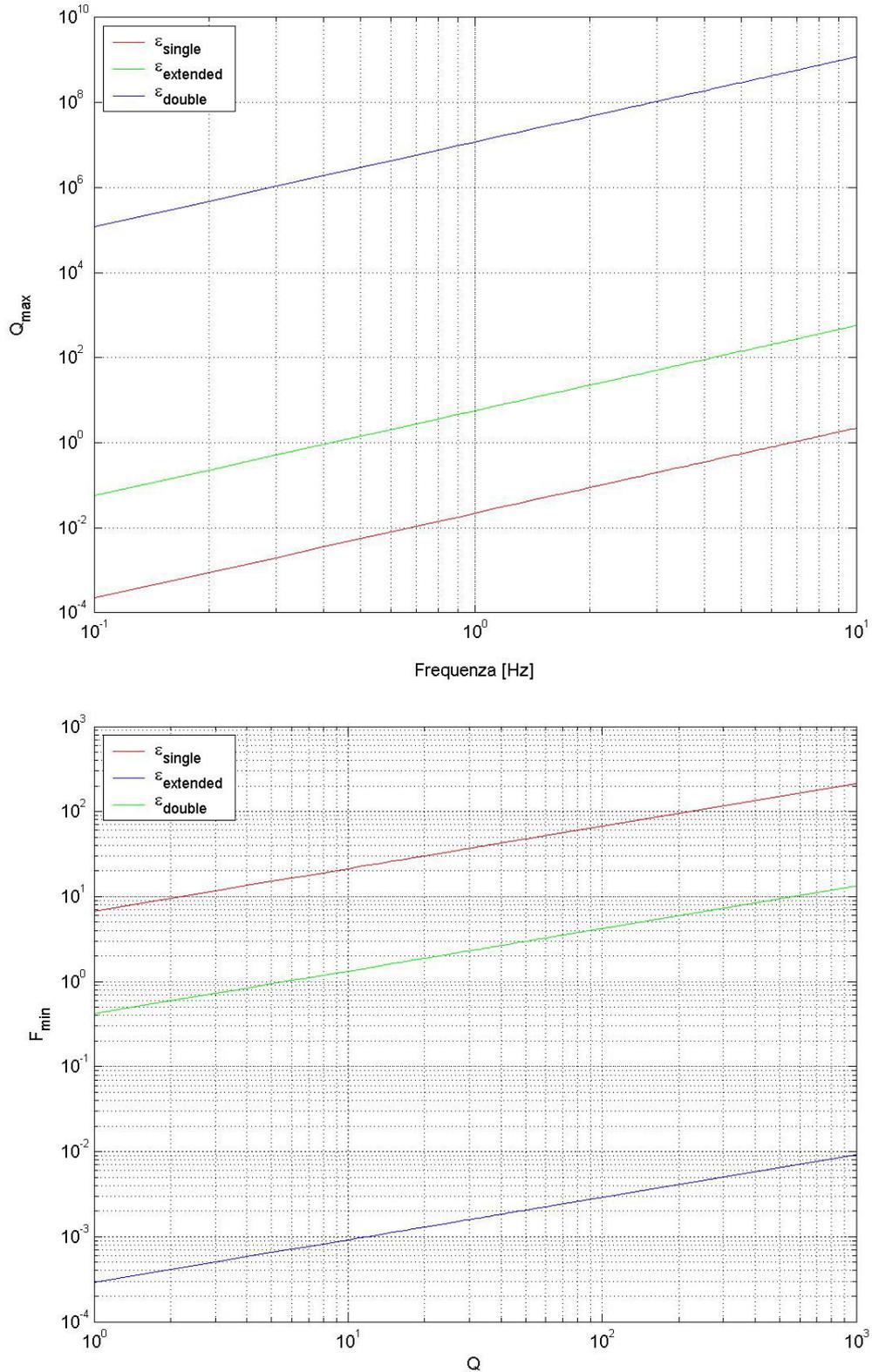


Figura 3.7: Filtri del II ordine al variare della precisione numerica ϵ : a) Q massimo implementabile in funzione della frequenza $\omega/2\pi$, b) Q minima implementabile in funzione di Q . In entrambi i casi è stato usato un fattore di sicurezza $S = 10$.

In Virgo c'è spesso la necessità di smorzare risonanze ad elevato Q in un range di frequenze ben al di sotto di 1 Hz. Tuttavia osservando le fig. 3.7 a) e b) che mostrano le diseguaglianze (3.77) e (3.76) con $S = 10$ per i vari formati floating point, notiamo che, anche usando la precisione singola estesa, è possibile implementare filtri solo con $Q \leq 5$ per $\omega/2\pi \simeq 1$ Hz. La soluzione di adottare un'aritmetica in precisione doppia non è realizzabile in quanto i tempi di elaborazione crescerebbero eccessivamente. Infatti come vedremo in 4.1 il processore usato nel controllo locale di Virgo, il Motorola DSP96002, appartiene ad una vasta categoria di processori, detti *DSP*, che posseggono un'architettura interna ottimizzata per l'elaborazione numerica dei segnali. In particolare nel DSP96002 è presente un'unità aritmetica (ALU) che consente, grazie a due unità di calcolo parallele, di eseguire, nel stesso ciclo di clock, somme e moltiplicazioni di numeri floating point in precisione singola estesa. Questa caratteristica non può essere sfruttata usando la precisione doppia ed i tempi di calcolo crescono di quasi un fattore 100.

3.2.7 Filtri del II ordine a bassa frequenza

Il metodo usato in Virgo, per risolvere il problema del limitato fattore Q dato dalla (3.77) senza ricorrere ad un incremento della precisione numerica, è quello di implementare i filtri del secondo ordine in bassa frequenza con due filtri del primo ordine in cascata. E' infatti sempre possibile esprimere la funzione di trasferimento di un filtro del II ordine come rapporto di polinomi di 1° grado in z a coefficienti complessi. Considerando un filtro di partenza con poli e zeri complessi coniugati, possiamo riscrivere $H(z)$ come

$$H(z) = B \frac{1 + az^{-1} + bz^{-2}}{1 + cz^{-1} + dz^{-2}} = B \frac{1 + uz^{-1}}{1 + vz^{-1}} \frac{1 + u^*z^{-1}}{1 + v^*z^{-1}} \quad (3.78)$$

dove $u, v \in \mathbb{C}$. Ponendo $u = a' + ib'$ e $v = c' + id'$ ed eguagliando i polinomi termine a termine abbiamo

$$\begin{aligned} 2a' &= a & a'^2 + b'^2 &= b \\ 2c' &= c & c'^2 + d'^2 &= d \end{aligned} \quad (3.79)$$

Da queste relazioni e dalla (3.78) è possibile ricavare

$$\begin{cases} a' = \frac{\omega^2 T^2 - 4}{4 + \frac{2\omega T}{Q} + \omega^2 T^2} \\ b' = \frac{2\omega T \sqrt{4 - \frac{1}{Q^2}}}{4 + \frac{2\omega T}{Q} + \omega^2 T^2} \end{cases} \quad (3.80)$$

Come prima è sufficiente studiare solo il numeratore di $H(z)$. Invertendo la (3.80) otteniamo

$$\begin{cases} \omega = \frac{2}{T} \sqrt{\frac{1+a'^2+2a'+b'^2}{1+a'^2-2a'+b'^2}} \\ Q = \frac{\sqrt{(1+a'^2+2a'+b'^2)(1+a'^2-2a'+b'^2)}}{2(1-a'^2-b'^2)} \end{cases} \quad (3.81)$$

Datoché risultava $a \simeq -2$ e $b \simeq 1$, si ha che $a' \simeq 1$ e $b' \simeq 0$. L'equazione (3.71) diviene

$$\begin{cases} \Delta\omega = \left| \frac{\partial\omega}{\partial a'} \right| \Delta a' + \left| \frac{\partial\omega}{\partial b'} \right| \Delta b' \\ \Delta Q = \left| \frac{\partial Q}{\partial a'} \right| \Delta a' + \left| \frac{\partial Q}{\partial b'} \right| \Delta b' \end{cases} \quad (3.82)$$

Calcolando le varie derivate parziali si ottiene:

$$\begin{aligned} \left(\frac{\partial\omega}{\partial a'} \right)_{\substack{a'=1 \\ b=0}} &= \frac{1}{2QT}; & \left(\frac{\partial\omega}{\partial b'} \right)_{\substack{a'=1 \\ b=0}} &= \frac{1}{2T} \sqrt{4 - \frac{1}{Q^2}} \\ \left(\frac{\partial Q}{\partial a'} \right)_{\substack{a'=1 \\ b=0}} &= \frac{1}{2\omega T}; & \left(\frac{\partial Q}{\partial b'} \right)_{\substack{a'=1 \\ b=0}} &= \frac{Q}{2\omega T} \sqrt{4 - \frac{1}{Q^2}} \end{aligned} \quad (3.83)$$

Abbiamo quindi

$$\begin{cases} \Delta\omega = \frac{\epsilon}{2QT} \\ \Delta Q = \frac{\epsilon}{2\omega T} \end{cases} \quad (3.84)$$

Imporre la condizione (3.75) non comporta più limitazioni al valore di Q . Inoltre il limite inferiore sulla frequenza viene notevolmente ridotto: l'equivalente della diseguaglianza vale

$$F > \frac{\epsilon f_c}{4\pi} \simeq \begin{cases} 95 \text{ } \mu\text{Hz} & \text{in precisione singola} \\ 0.4 \text{ } \mu\text{Hz} & \text{in precisione singola estesa} \end{cases} \quad (3.85)$$

E' quindi ora possibile implementare, usando la precisione estesa, risonanze nel range del μHz con fattori di qualità arbitrariamente elevati. Tutti questi risultati possono essere estesi anche al caso in cui $H(z)$ possieda zeri o poli reali.

In termini di tempo di elaborazione, il prezzo da pagare per questa realizzazione alternativa dei filtri del II ordine è legato all'uso dei coefficienti complessi. Si deve infatti rappresentare separatamente la parte reale e la parte immaginaria di ogni coefficiente incrementando il numero di accessi alla memoria del calcolatore. La realizzazione di un filtro reale del secondo ordine comporta, vedi (3.47), 8 accessi alla memoria e l'esecuzione di 4 somme e 5 prodotti. Si può dimostrare, considerando il sistema di equazioni alle differenze corrispondente alla (3.78), che per l'equivalente cascata

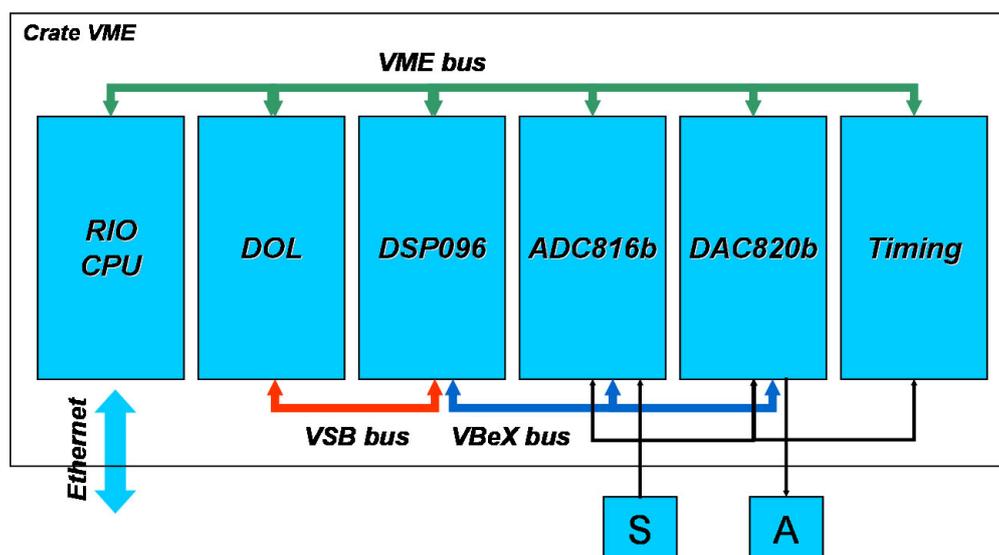


Figura 3.8: Schema a blocchi dell'elettronica del controllo locale

di filtri complessi del primo ordine sono necessari 12 accessi alla memoria e l'esecuzione di 13 somme e 14 prodotti. Questo si traduce in un incremento del tempo di calcolo di un fattore 2, valore che giustifica la scelta di questa implementazione rispetto all'uso della precisione doppia.

3.3 L'attuale sistema di controllo

Lo schema a blocchi generale dell'attuale sistema di controllo locale è mostrato in fig. 3.8.

Descriviamo le parti che lo costituiscono:

3.3.1 Scheda ADC816b

E' una scheda a 8 canali basata sul convertitore analogico digitale a 16 bit AD976A della Analog Devices. Esso assicura un flusso dati massimo di 200 kSPS (SPS = campioni/secondo). Lo schema a blocchi del convertitore è mostrato in fig. 3.9 [16]. Quando i segnali di controllo \overline{CS} ed R/\overline{C} vengono mantenuti ad un livello basso per più di 50 ns, inizia la conversione e l'ingresso analogico giunge ad un ADC ad approssimazioni successive (SAR) a redistribuzione di carica. Quest'ultimo è connesso ad un'interfaccia parallela ad alta velocità e ad un circuito logico di controllo e calibrazione.

Lo schema elettrico tipico di un ADC SAR a 5 bit, basato sulla redistribuzione di carica, è mostrato nelle fig. 3.10 a)-c) [17]. Tutti i condensatori hanno valori di capacità binari ($C, C/2, \dots, C/2^{n-1}$ dove n è il numero di

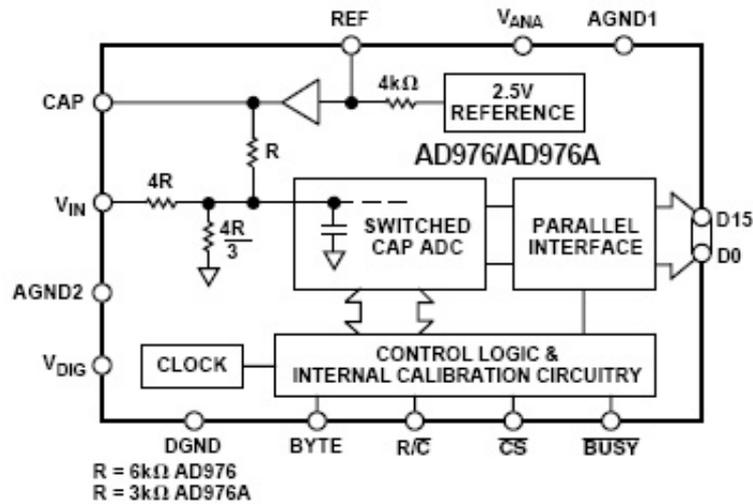


Figura 3.9: Schema a blocchi dell'ADC AD976A

bit del convertitore) pesati per ognuno degli $n + 1$ switch. Gli ultimi due condensatori hanno valori identici $C/2^{n-1}$ in modo da ottenere una capacità totale $2C$. Gli switch sono implementati attraverso transistor MOS. La tensione di riferimento V_{ref} , nel caso dell'ADC976A, è generata internamente e vale $V_{ref} = 2.5$ V.

Nel primo passo della conversione (fig. 3.10 a)), il condensatore di capacità C viene connesso a V_{ref} mentre gli altri sono posti a massa. Quindi la tensione all'ingresso del comparatore vale $V_c = -V_{IN} + V_{ref}/2$. A seconda che sia $V_{IN} > V_{ref}/2$ o $V_{IN} < V_{ref}/2$ l'uscita del comparatore è alta o bassa e si ottiene il bit MSB. Nel secondo passo il condensatore di capacità $C/2$ viene connesso a V_{ref} . Se MSB= 0 lo switch corrispondente viene posto a massa (fig. 3.10 b)) e l'ingresso del comparatore vale $V_c = -V_{IN} + V_{ref}/4$. Se MSB= 1 lo switch rimane connesso a V_{ref} (fig. 3.10 c)) e $V_c = -V_{IN} + 3/4 \cdot V_{ref}$. Il processo di conversione procede in questo modo per n passi finché tutti i bit sono stati generati e $V_c = -V_{IN} + \sum_{i=1}^n x_i \frac{V_{ref}}{2^{n-i}}$ dove x_i è il valore del bit i . Il numero binario ottenuto è in complemento a due. Per incrementare l'accuratezza della conversione il circuito di controllo utilizza una serie di coefficienti di calibrazione memorizzati in una ROM resistiva.

Ciascuna ADC della scheda ha ingresso un proprio amplificatore in serie ad un filtro di *antialiasing* ellittico del 7° ordine con frequenza di taglio 3.8 kHz e impedenza d'ingresso 100 kΩ.

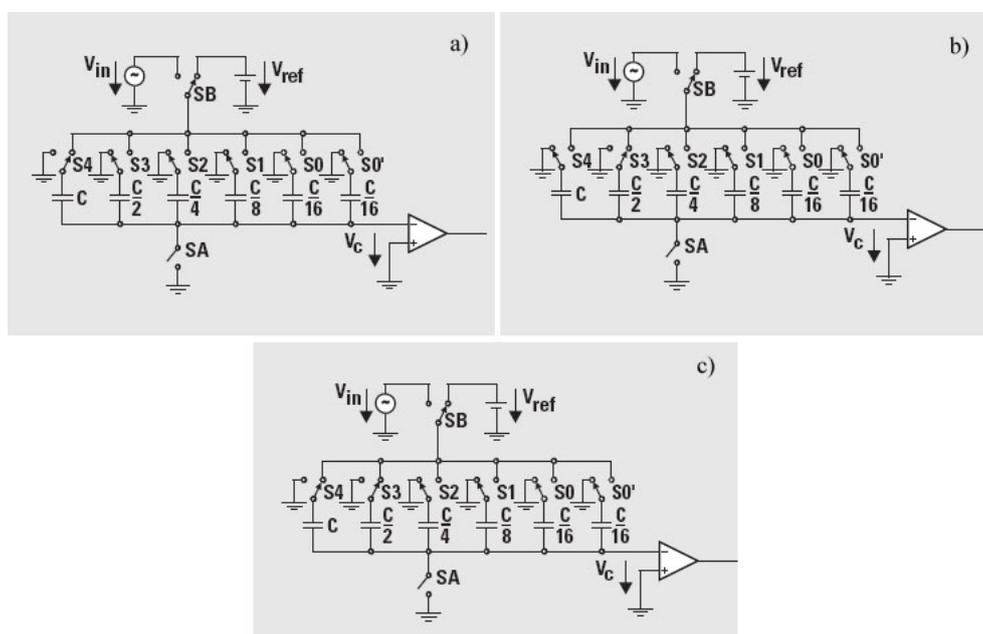


Figura 3.10: Fasi di conversione di un tipico ADC SAR a redistribuzione di carica

3.3.2 Scheda DAC820b

E' una scheda ad 8 canali sviluppata dall'INFN di Pisa che si interfaccia con i due principali bus usati in Virgo, il VME ed il VBeX. La DAC820b è basata sul convertitore digitale analogico a 20 bit AD1862 della Analog Devices. Progettato per applicazioni audio, consente un rapporto segnale rumore di 120 dB ed una ridotta distorsione armonica. Accetta in input parole a 20 bit in complemento a 2, supportando un flusso dati massimo di 17Mbit. Ciascuno degli 8 DAC ha in uscita un filtro di ricostruzione ellittico del 7° ordine con frequenza di taglio 3.7 kHz. Il segnale d'uscita ha un range di ± 5 V.

Il ritardo introdotto dalla scheda nel ciclo di controllo è pari alla somma del tempo di conversione del DAC con il tempo richiesto per accedere alla scheda. Il primo vale $1.75 \mu\text{s}$ mentre il secondo, sia che si usi il bus VME che il VBeX, vale circa 200 ns. Poiché per scrivere 20 bit sono necessari 2 accessi $t_{delay} \simeq 2.15 \mu\text{s}$ per un singolo canale. Il flusso dati massimo è quindi 460 kHz per 1 canale e 200 kHz per tutti gli 8 canali.

3.3.3 Scheda DSP

La scheda DSP è il cuore del sistema di controllo locale. Il compito della scheda DSP è il calcolo di filtri digitali IIR del I e del II ordine e l'esecuzione

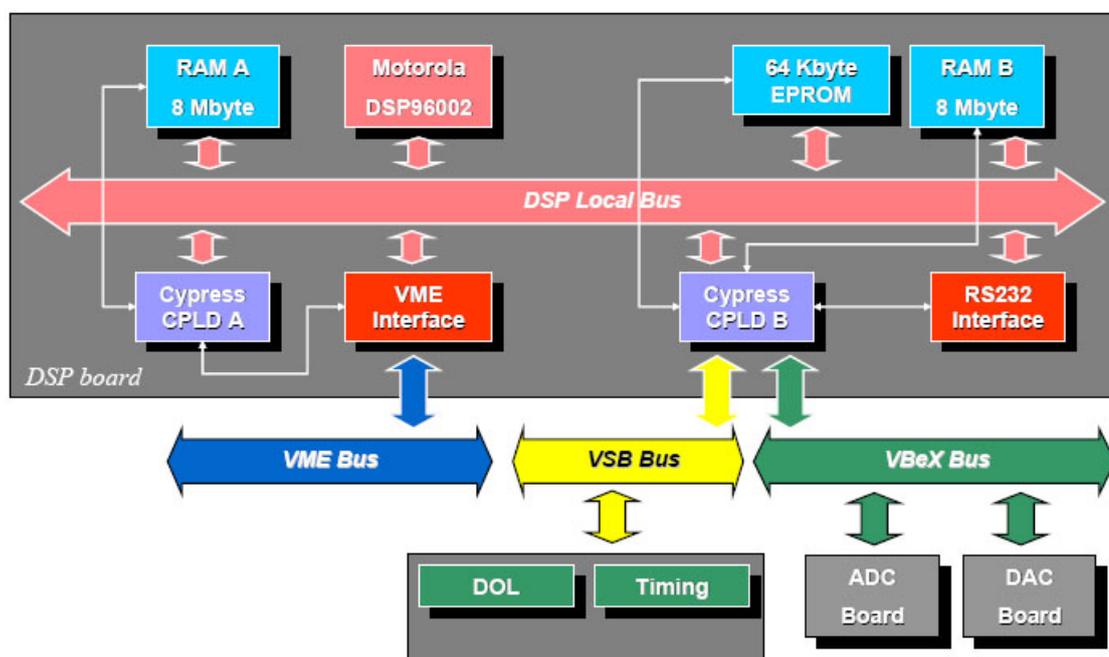


Figura 3.11: Schema a blocchi della scheda DSP

di altre operazioni al fine di smorzare le risonanze presenti nel moto del SA. Attualmente la scheda DSP effettua calcoli equivalenti all'implementazione di 640 poli e zeri. Come vedremo, tali calcoli devono essere completati entro un tempo massimo di $80 \mu s$.

Sviluppata dall'INFN di Pisa nel 1996, la scheda DSP si interfaccia con tutti i bus standard usati in Virgo: VME, VSB, VBeX [18]. Lo schema a blocchi è mostrato in fig. 3.11. Analizziamone i componenti principali:

Motorola DSP96002 È stato il primo DSP a virgola mobile disponibile sul mercato. È in grado di eseguire operazioni aritmetiche con quattro formati numerici: due formati a virgola fissa a 32 bit (uno in complemento a due, l'altro in *sign and magnitude*), il formato floating point in precisione singola previsto dallo standard IEEE/ANSI 754-1985, un formato floating point in precisione singola estesa con 11 bit di esponente e 32 bit di mantissa [19]. Nella versione usata in Virgo con clock a 60 MHz è capace di eseguire fino a 30 milioni di istruzioni per secondo (MIPS) ed un massimo di 90 milioni di operazioni in virgola mobile al secondo (MFLOPS). I tempi di calcolo ottenibili dal DSP96002 in alcuni algoritmi tipici dell'elaborazione digitale dei segnali sono riassunti nella seguente tabella:

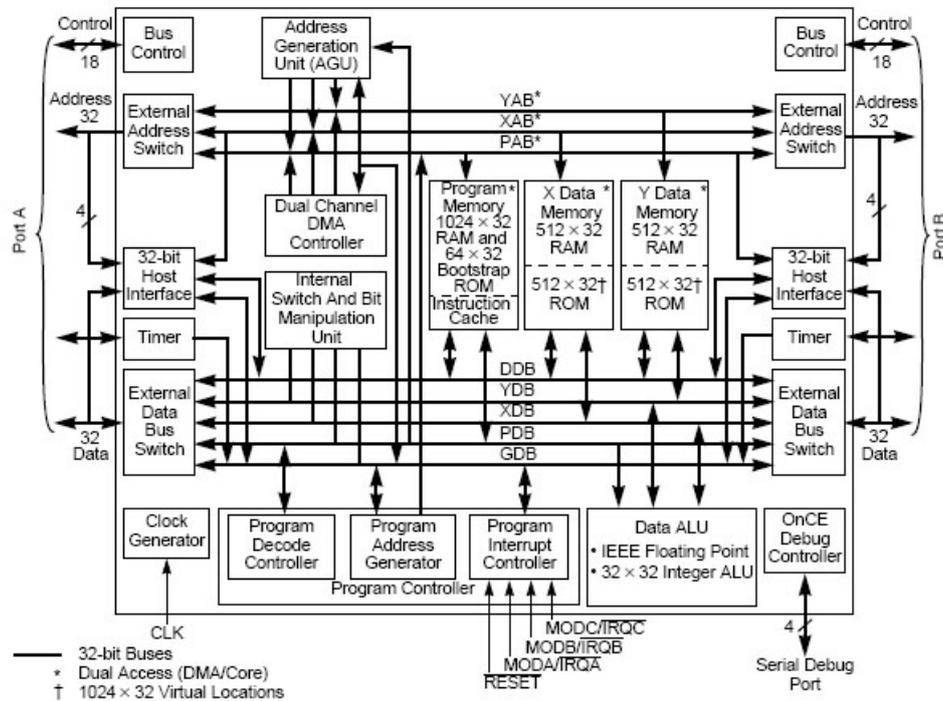


Figura 3.12: Schema a blocchi dell'architettura interna del Motorola DSP96002

Algoritmo	t_{CPU}
FFT complessa a 1024 punti	700 μ s
Filtro FIR (per tap)	34 ns
Filtro IIR del II ordine	266 ns
Prodotto matriciale	
$[3 \times 3] \times [3 \times 1]$	400 ns
$[4 \times 4] \times [4 \times 1]$	634 ns
Divisione (y/x)	232 ns

Lo schema a blocchi della sua architettura interna è mostrato in fig. 3.12. I costituenti principali sono

- **Bus dati:** i movimenti di dati internamente al processore avvengono attraverso 5 bus bidirezionali a 32 bit: X Data Bus (XDB), Y Data Bus (YDB), Global Data Bus (GDB), DMA Data Bus (DDB) e Program Data Bus (PDB). I bus dati X ed Y possono essere trattati come un unico bus a 64 bit. Tutti i trasferimenti dalla ALU alle memorie dati X ed Y avvengono tramite XDB ed YDB, mentre il caricamento delle istruzioni avviene tramite PDB.

- **Memoria programma:** è costituita da una memoria RAM con 1024 x 32 bit possibili allocazioni e da una memoria ROM di 64 x 32 bit usata per l'inizializzazione del processore detta *bootstrap*. La memoria programma può contenere istruzioni, costanti, matrici fissate al momento della compilazione del codice.
- **Memorie dati X e Y:** sono entrambe divise in RAM dati, costituita dalle 512 x 32 bit allocazioni più basse dello spazio di memoria dati, e ROM dati, con 1024 x 32 bit possibili allocazioni. Gli indirizzi generati dalla AGU giungono alle memorie X ed Y attraverso i corrispondenti bus XAB ed YAB. I trasferimenti dati avvengono tramite i bus XDB ed YDB.
- **Unità aritmetica (ALU):** esegue tutte le operazioni logiche ed aritmetiche sugli operandi. Essa è costituita da 10 registri a 96 bit, un sommatore a 32 bit, un moltiplicatore a 32 bit. I dati vengono memorizzati nei registri nel formato a virgola mobile in precisione doppia. Ciascun registro della ALU può essere letto o scritto dai XDB e YDB come operando a 32 o 64 bit. Come già accennato, la ALU supporta due formati a 32 bit a virgola fissa, e due formati a virgola mobile, uno in precisione singola standard ed uno in precisione estesa a 44 bit. Tutti calcoli vengono eseguiti in precisione estesa ed eventualmente arrotondati in precisione singola.
- **Unità di generazione degli indirizzi (AGU):** è responsabile dell'archiviazione e dell'effettivo calcolo degli indirizzi degli operandi ed è usato sia dal controllore della memoria interna integrato nel processore (*DMA controller*) che dal processore stesso. L'AGU contiene un totale di 24 registri a 32 bit. Ciascun registro può essere letto dai bus XAB, YAB, PAB e letto o scritto dal bus globale GDB. L'AGU può generare due indirizzi a 32 bit ogni singolo ciclo di clock.

Nel capitolo successivo vedremo come molte delle caratteristiche sopra elencate, in particolare la separazione tra memoria dati e memoria programma, differenzino la famiglia dei DSP dai processori tradizionali, determinandone le superiori prestazioni nel calcolo numerico.

Interfaccia VME L'interfaccia VME è implementata attraverso il VIC068A, un chip programmabile prodotto dalla Cypress Semiconductor che facilita l'interfacciamento del bus locale di un microprocessore generico con il bus VME.

Memorie Vengono utilizzati due banchi di memoria RAM statica di 2048 K x 32 bit ciascuno ed una EPROM AMD27512 da 64 KB. Quest'ultima viene usata esclusivamente per l'inizializzazione del processore.

CPLDA e CPLDB: Sulla scheda sono installate due logiche programmabili CPLD (Complex Programmable Logic Device), modello CY7C374, prodotte dalla Cypress. La CPLDA si occupa della gestione di uno dei due banchi di memoria statica e dell'arbitraggio dei segnali di controllo del VIC068A. Compito della CPLDB è invece interfacciare i bus VSB e VBeX con il bus locale del DSP e gestire le due porte seriali, la EPROM, e l'altro banco di memoria statica. Il software necessario per implementare queste funzioni è stato sviluppato in linguaggio VHDL (v. cap. 5).

3.3.4 RIO8062

La CES RIO8062 è la scheda CPU standard usata in Virgo. Basata sul processore PowerPC 604e (vedi par. 4.2.1) a 400 MHz della Motorola, consente di ottenere un flusso dati massimo di 60 MB/s sul bus VME e di 30MB/s sul bus VME [20].

Il sistema operativo usato, il LynxOS prodotto dalla LinuxWorks, è uno UNIX progettato espressamente per applicazioni in tempo reale di tipo *hard* (vedi par. 4.1) quali sono il controllo globale e il controllo locale di Virgo. Infatti il kernel di LynxOS è stato sviluppato in modo da rendere le routine di gestione degli interrupts molto brevi e veloci. Questa caratteristica consente un pieno determinismo dei tempi di risposta del sistema operativo indipendentemente dal flusso dati e dal carico del processore.

3.3.5 *Digital Optical Link* (DOL)

I vari sottosistemi dell'esperimento (i fotodiodi di lettura, il controllo globale, il controllo locale, il centro di acquisizione dati detto *Frame Builder*) scambiano segnali digitali attraverso fibre ottiche connesse ad una scheda elettronica, compatibile con gli standard VME e VSB, detta DOL (Digital Optical Link), realizzata dal laboratorio LAPP di Annecy [21]. Essa è capace di inviare piccole quantità di dati (alcuni KB) in modalità punto-punto fino a 3.2 Km di distanza senza introdurre alcun ritardo. Due canali completamente indipendenti si occupano della ricezione e dell'invio dei dati. Su ciascun canale una memoria FIFO gestisce il funzionamento asincrono della connessione ottica con il bus locale della scheda DOL. La FIFO connessa con il canale di ricezione possiede 512 x 16 bit allocazioni mentre quella dedicata al canale di trasmissione è di 512 x 32 bit.

3.3.6 *Timing*

Ai fini del corretto funzionamento dell'intero strumento è necessario che tutti i cicli di controllo funzionino in modo sincrono. Per questo in Virgo è presente un segnale di clock centralizzato che usa come riferimento un sistema GPS installato nell'edificio di controllo. Attraverso connessioni in fibra

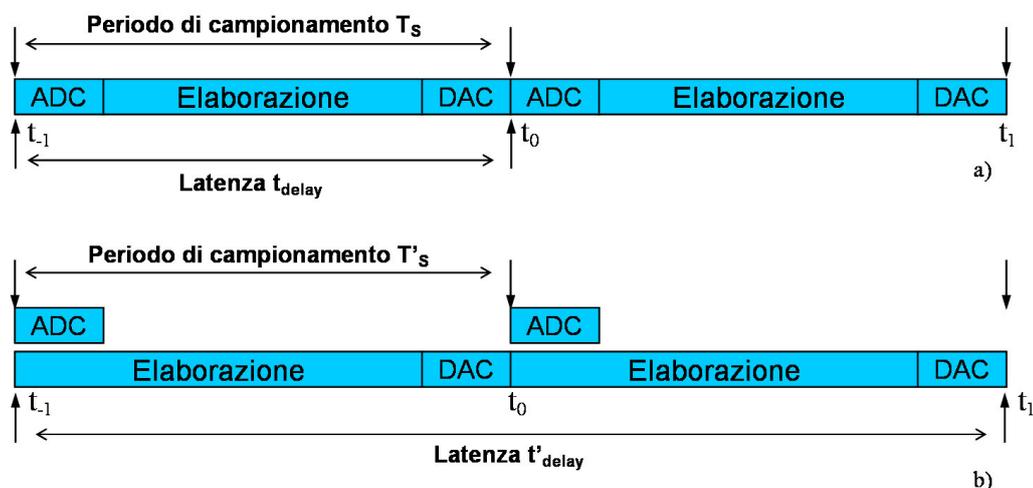


Figura 3.13: I due possibili modelli di sincronizzazione per un sistema di controllo digitale

ottica, questo segnale viene trasferito a tutti i sottosistemi dell'esperimento. La funzione della scheda di Timing è di convertire localmente il clock in segnali TTL ed interrupt del bus VME in modo da poter essere usato dall'elettronica di controllo. La scheda è dotata inoltre di un oscillatore integrato che consente di utilizzarla anche in assenza del clock centrale.

3.3.7 Sincronizzazione

La sincronizzazione è uno degli aspetti chiave di ogni sistema di controllo in retroazione digitale. E' necessario garantire che i ritardi introdotti dalle varie componenti dell'elettronica non introducano un eccessivo sfasamento nella banda di controllo. Tale sfasamento si traduce infatti in una riduzione del margine di fase M_F del sistema compromettendone la stabilità.

Le principali fonti di ritardo sono i tempi di conversione dell'ADC e del DAC, gli sfasamenti introdotti dai filtri analogici di *antialiasing* e di ricostruzione, il tempo di elaborazione del calcolatore.

Il requisito principale è che il sistema funzioni in modo sincrono ossia che il ritardo tra l'ingresso e l'uscita, detto latenza, sia indipendente dal tempo di elaborazione. A tal fine sono possibili i due possibili approcci mostrati in fig. 3.13 a) e b).

Nel primo caso all'istante t_0 inizia la conversione analogico-digitale. Viene quindi inviato un segnale di *interrupt* al processore che inizia l'elaborazione dell'ultima sequenza numerica acquisita. Infine il dato elaborato giunge

in ingresso al DAC che lo riconverte in forma analogica. In definitiva il periodo di sampling $T_s = \frac{1}{f_c} = t_{ADC} + t_{CPU} + t_{DAC}$ e la latenza $t_{delay} = T_s$.

Usando il secondo approccio la conversione analogico-digitale e l'elaborazione iniziano nello stesso istante t_0 . In questo modo il processore filtra la sequenza acquisita all'istante t_{-1} e risulta $T'_s = t_{DAC} + t_{CPU}$. Questo aumento della frequenza di campionamento si ottiene a discapito della latenza $t'_{delay} = 2T'_s$. Al crescere di f_c questa soluzione risulta molto conveniente in quanto mentre T_s diventa trascurabile, t_{ADC} rimane un tempo morto costante nell'elaborazione dei dati. Questo schema di sincronizzazione è quello adottato in Virgo.

Possiamo adesso stimare il valore dei vari contributi al ritardo.

Filtri analogici: come già accennato i filtri di *antialiasing* e di ricostruzione usati in Virgo sono filtri ellittici del 7° ordine. I filtri di questo tipo sono caratterizzati dalla proprietà

$$|H(j\omega)|^2 = \frac{1}{1 + \epsilon^2 U_n^2(\omega)} \quad (3.86)$$

dove ϵ è una costante e le U_n sono dette funzioni ellittiche di Jacobi. Ai fini del calcolo del ritardo introdotto possiamo approssimare il comportamento del filtro ellittico nella banda passante con un filtro di Butterworth dello stesso ordine. Nei filtri di Butterworth la relazione (3.86) diventa

$$|H(j\omega)_n|^2 = \frac{1}{1 + (j\omega/j\omega_0)^{2n}} \quad (3.87)$$

dove ω_0 è la pulsazione corrispondente alla frequenza di taglio ed n è l'ordine del filtro. Usando questa proprietà è possibile scrivere la funzione di trasferimento che assume la forma

$$H_n(s) = \omega_0^n \prod_{k=1}^n \frac{1}{s - \omega_0 p_k} \quad (3.88)$$

dove p_k sono le radici di polinomi di ordine n detti polinomi di Butterworth. I poli di $H_n(s)$, rappresentati nel piano s , sono distribuiti uniformemente, a distanza angolare π/n , su una circonferenza di raggio ω_0 in posizione simmetrica rispetto all'asse reale (vedi fig. 3.14).

A questo punto il ritardo t_{AA} può essere stimato scomponendo il filtro del 7° ordine con una serie di filtri in cascata del I e del II ordine, analogamente a quanto fatto in precedenza. Si definisce ritardo di gruppo la quantità

$$\tau = \lim_{\omega \rightarrow 0} \frac{\partial \phi}{\partial \omega} \quad (3.89)$$

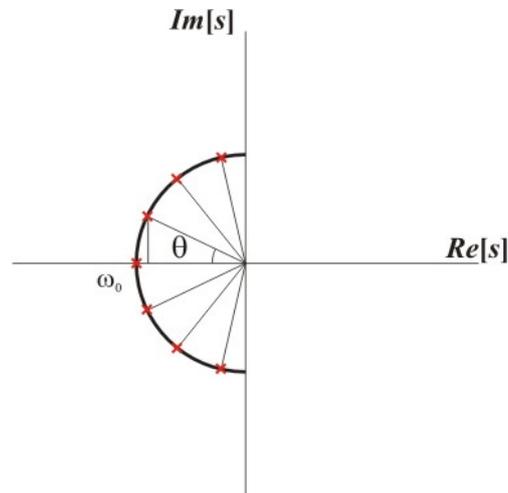


Figura 3.14: Rappresentazione nel piano s dei poli di un filtro di Butterworth del 7° ordine

dove ϕ è lo sfasamento introdotto dal filtro. Nel caso di un filtro di Butterworth del I ordine la funzione di trasferimento diventa

$$H_1(s) = \frac{1}{s/\omega_0 + 1}$$

Quindi lo sfasamento risulta

$$\phi_1 = -\arctan \frac{\text{Im}[H_1(s)]}{\text{Re}[H_1(s)]} = -\arctan(\omega/\omega_0)$$

e il ritardo vale

$$\tau_1 = \lim_{\omega \rightarrow 0} \frac{\partial \phi_1}{\partial \omega} \simeq \frac{1}{\omega_0}$$

Effettuando lo stesso calcolo effettuato per un generico filtro del II ordine otteniamo

$$\tau_2 = -\frac{1}{Q\omega_0} \quad (3.90)$$

dove Q è il fattore di qualità della risonanza. Da ciò emerge che la presenza di zeri nella funzione di trasferimento di un filtro non ha alcun effetto sul ritardo introdotto. Nel caso di coppie di poli simmetriche rispetto all'asse reale il fattore di qualità è semplicemente

$$Q = \frac{1}{2 \cos \theta} \quad (3.91)$$

dove θ è l'angolo che i poli formano con l'asse reale (v. fig. 3.14). Datoché si può dimostrare che il ritardo introdotto da una serie di

sistemi in cascata è equivalente alla somma dei singoli ritardi, abbiamo

$$t_{AA} \simeq \frac{1}{\omega_0} \left(2 \cos \frac{\pi}{7} + 2 \cos \frac{2\pi}{7} + 2 \cos \frac{3\pi}{7} + 1 \right) \simeq \frac{4.5}{\omega_0} \quad (3.92)$$

Datoché, come già accennato nel cap. 2, la frequenza di campionamento dei controlli di Virgo è 10 kHz e la frequenza di taglio $\omega_0/2\pi$ dei filtri analogici vale 3.8 kHz, possiamo esprimere il ritardo in funzione del periodo di campionamento

$$t_{AA} \simeq 1.9T_s \quad (3.93)$$

Poiché le frequenze di taglio del filtro di antialiasing e di ricostruzione sono circa uguali il ritardo complessivo introdotto è circa $3.8T_s$.

DAC: I contributi principali sono il tempo di *hold* t_{hold} e di *settling* t_{st} . Il tempo di *hold* è il tempo per cui il DAC mantiene fisso l'ultimo valore analogico dell'uscita prima dell'arrivo del campione successivo. Esso equivale a $t_{hold} = \frac{1}{2}T_s$.

Il tempo di *settling* corrisponde al tempo che l'uscita del DAC impiega per raggiungere il suo valore finale misurato dall'istante di transizione dell'impulso digitale. Nel caso dell'AD1862 t_{st} vale 350 ns e può dunque essere trascurato. Abbiamo quindi:

$$t_{DAC} = t_{hold} = \frac{1}{2}T_s \quad (3.94)$$

Combinando tutti i contributi otteniamo infine il ritardo totale:

$$t_{delay} = 2t_{AA} + t_{CPU} + t_{DAC} = 6.3T_s \quad (3.95)$$

3.3.8 Limiti del sistema

Noto il ritardo complessivo t_{delay} introdotto dall'elettronica, è possibile, utilizzando la definizione di M_F data dalla (3.24), ottenere una condizione sulla minima frequenza di campionamento f_c richiesta dal nostro sistema. Infatti, supponendo un comportamento della funzione di trasferimento come $1/f$ alla frequenza di guadagno unitario ed utilizzando un margine di fase $M_F = 45^\circ$, anche lo sfasamento introdotto dal ritardo alla frequenza f_U di guadagno unitario deve essere pari a 45° . Vale quindi

$$\frac{1}{8t_{delay}} \simeq \frac{f_c}{48} = f_U \quad (3.96)$$

E' quindi necessario avere una stima della frequenza di guadagno unitario del controllo globale. Tale stima può essere ricavata dal tempo che i controlli di

Virgo hanno a disposizione per porre le due cavità Fabry-Perot dell'interferometro in risonanza con il laser. Questo tempo è dato approssimativamente dal rapporto tra la larghezza a metà altezza della risonanza, pari a $\frac{\lambda}{4\mathfrak{F}}$ (vedi cap. 2), e la velocità v_s dello specchio. Otteniamo quindi

$$\frac{1}{f_U} \simeq \frac{\lambda}{4\mathfrak{F}v_s} \quad (3.97)$$

A circuito aperto infatti, ossia senza l'intervento dei controlli dell'interferometro, il moto dello specchio causato dal rumore sismico è caratterizzato da una velocità v_s pressoché costante. Tale velocità varia, a seconda delle condizioni sperimentali, in un intervallo $1 \mu\text{m/s} < v_s < 10 \mu\text{m/s}$. Data la variabilità di v_s , f_U non ha un valore unico ma è compresa nell'intervallo $200 \text{ Hz} < f_U < 2 \text{ kHz}$. Quindi a seconda delle condizioni sperimentali la f_c minima richiesta per un margine di fase di 45° è compresa nell'intervallo

$$10\text{kHz} < f_c < 100\text{kHz} \quad (3.98)$$

Nel sistema attuale la frequenza di campionamento è pari a 10 kHz ossia al valore minimo dell'intervallo. E' quindi necessario trovare un difficile compromesso tra il margine di fase, che deve essere sufficientemente elevato da garantire la stabilità, e la frequenza f_U , che in condizioni sfavorevoli può, come abbiamo detto, raggiungere valori di 2 kHz. Questo si traduce nell'impossibilità di mantenere l'interferometro nel punto di lavoro nelle condizioni sperimentali più sfavorevoli in termini di rumore sismico. L'incremento della frequenza f_c non è una soluzione adottabile in quanto causerebbe una riduzione del tempo di elaborazione t_{CPU} non sostenibile dalla potenza di calcolo del DSP96002. Infatti, come abbiamo già detto, il processore deve eseguire calcoli per un'equivalente di 640 poli e zeri in un tempo pari a circa l'80% del periodo di campionamento $1/f_c$. Quindi, utilizzando i valori dei tempi di esecuzione forniti dalla tabella del par. 3.3.3, vediamo che l'attuale tempo di calcolo è già molto vicino al limite massimo.

L'incremento della frequenza di campionamento risulta dunque impossibile senza un aumento della potenza di calcolo della scheda DSP. Nasce così l'esigenza di progettare un nuovo circuito dedicato al controllo delle sospensioni di Virgo che costituisca l'evoluzione di quello attuale. Lo scopo di questa tesi specialistica è appunto la progettazione di tale circuito.

Capitolo 4

I DSP

Dopo aver introdotto il concetto di elaborazione in tempo reale, si effettua un confronto tra i processori dedicati all'elaborazione dei segnali (DSP) e le altre famiglie di calcolatori attualmente disponibili. Si descrivono quindi le caratteristiche dell'ADSP21160N, l'unità scelta per il progetto del circuito.

4.1 Elaborazione in tempo reale

Tutte le svariate applicazioni in cui vengono usati gli elaboratori possono essere essenzialmente divise in due categorie:

- la manipolazione dei dati
- il calcolo numerico

Rientrano nella prima categoria tutte le applicazioni che richiedono un frequente uso dei test condizionali e numerosi trasferimenti di dati da un'area e all'altra della memoria del calcolatore. Esempi tipici sono la gestione dei *database*, l'elaborazione dei testi, i sistemi operativi.

Il principale fattore determinante nel calcolo numerico è invece il numero di operazione al secondo che il processore riesce ad eseguire nei vari formati di rappresentazione numerica descritti nel precedente capitolo. Lo sviluppo di filtri dedicati a sistemi di controllo appartiene a questa categoria di applicazioni. Se analizziamo infatti l'implementazione di un filtro digitale di ordine N , descritta dal sistema (3.47), notiamo che il tempo di elaborazione è determinato dai tempi di accesso alla memoria e di esecuzione delle somme e dei prodotti. Inoltre la discussione del par. 3.3.4 ha mostrato l'importanza del rispetto dei vincoli temporali per non compromettere la stabilità del sistema. E' quindi fondamentale che il processore usato in un sistema di controllo abbia dei tempi di elaborazione prevedibili in modo esatto.

Emerge il concetto di sistema di controllo in tempo reale ossia un sistema in cui il segnale d'uscita viene prodotto contemporaneamente all'acquisizione

del segnale in ingresso. Un processo in tempo reale deve quindi terminare entro una scadenza temporale fissata, dipendente dall'ambiente in cui esso opera, detta *deadline*. Si distinguono inoltre due classi di sistemi in tempo reale [22]:

- sistemi in tempo reale *hard*
- sistemi in tempo reale *soft*

Mentre per la seconda categoria una violazione della *deadline* non ha conseguenze critiche sul sistema di controllo, per la prima può compromettere la stabilità e quindi il corretto funzionamento del sistema. Il controllo locale di Virgo appartiene a quest'ultima classe di controlli in tempo reale.

4.2 Architettura dei processori

4.2.1 La famiglia Intel x86

I processori della famiglia Intel x86 sono i più comunemente usati nei calcolatori commerciali. Essi sono caratterizzati da un'architettura, detta Von Neumann, che prevede la presenza di unico bus, il bus dati, attraverso il quale accedere sia ai dati che alle istruzioni, contenuti entrambi in un'unica area di memoria condivisa. Questa configurazione determina inevitabilmente un collo di bottiglia nell'elaborazione: infatti l'istruzione da decodificare e, successivamente, i dati sui quali operare passano attraverso lo stesso canale. L'esecuzione di una qualsiasi operazione aritmetica richiede quindi almeno tre cicli di clock, uno per trasferire ciascuno dei tre numeri dalla memoria al processore, attraverso il bus.

Per cercare di risolvere questo problema nelle più recenti generazioni di x86 è stato introdotto un meccanismo denominato *prefetch*: le istruzioni sequenzialmente successive a quella in esecuzione vengono inserite in una apposita coda interna al processore. Nel caso di istruzioni non sequenziali come i salti o le subroutine si usa la cosiddetta *branch prediction*: attraverso alcuni algoritmi specifici vengono determinate le istruzioni che hanno un'elevata probabilità di essere eseguite a breve e vengono portate all'interno del processore rendendole disponibili. Allo stesso tempo, per incrementare ulteriormente la velocità di accesso alle istruzioni, vengono utilizzate delle memorie di ridotta capacità, dette *cache*, che si interfacciano con il processore con una banda molto larga. Questi vari meccanismi, riassunti nella fig. 4.1 che rappresenta alcuni componenti dell'architettura di un processore Pentium, non sono tuttavia sufficienti a garantire delle prestazioni elevate nelle calcolo numerico. Infatti i processori Intel x86 sono progettati per applicazioni generiche che rientrano nell'ambito della manipolazione dei dati. Inoltre la natura stessa dei meccanismi predittivi usati non consente di

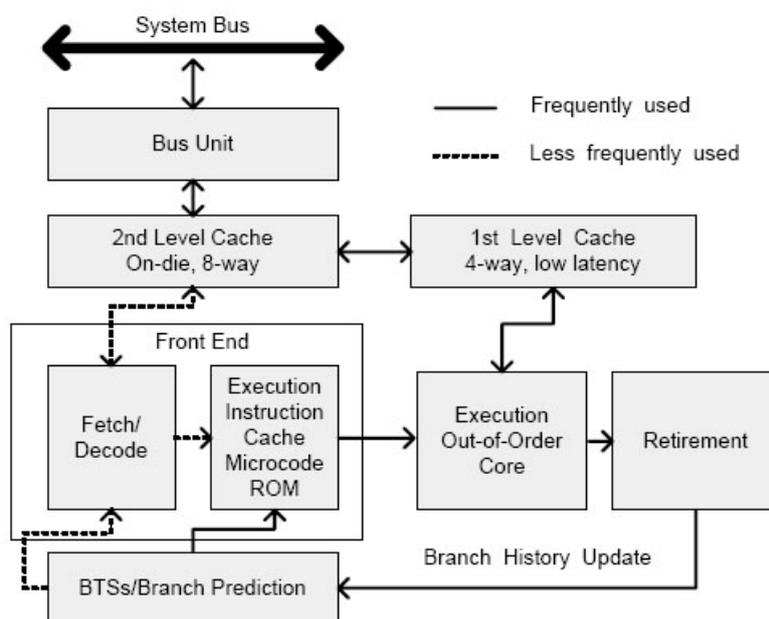


Figura 4.1: Alcuni dettagli dell'architettura di un processore Intel Pentium

prevedere l'esatto tempo di esecuzione di un algoritmo, caratteristica fondamentale per i sistemi in tempo reale. Sarebbe quindi impossibile realizzare un sistema di controllo utilizzando un processore della famiglia x86.

4.2.2 I RISC

L'acronimo RISC (Reduced Instruction Set Computer) indica un processore la cui architettura interna favorisce l'implementazione di un insieme ridotto di istruzioni caratterizzate da un tempo di esecuzione pressoché equivalente. In questi elaboratori viene eliminata l'unità di decodifica delle istruzioni che nei processori CISC (Complex Instruction Set Computer) come gli x86, provvede ad interpretare i comandi e ad inviare alla cpu dati e segnali necessari all'esecuzione. Nei processori RISC, per ovviare al problema che si verifica quando un'istruzione ha come operando il risultato di un'operazione precedente non ancora terminata, vengono adottate due diverse soluzioni:

- architettura SPARC (Scalable Processor ARChitecture): Viene utilizzato un meccanismo denominato interlock. Esso consiste nel bloccare, nel momento in cui un'istruzione viene immessa nella pipeline, il registro in cui essa andrà a scrivere il risultato. In questo modo, se un'altra istruzione deve leggere il contenuto di quel registro, viene fermata all'interno della pipeline fino a quando non viene scritto il va-

lore corretto nel registro stesso. Una variante di questa architettura è utilizzata nella famiglia dei processori PowerPC.

- architettura MIPS (Microprocessor without Interlocked Pipeline Stages): Viene eliminato il meccanismo di interlock e ridotto ulteriormente il numero di istruzioni disponibili.

Sebbene l'architettura dei processori RISC consenta di ottenere prestazioni più elevate degli x86 nel calcolo numerico, non permette una piena prevedibilità dei tempi di calcolo e non è quindi adeguata alla nostra applicazione.

4.2.3 Architetture Harvard e SHARC

Il principio base dell'architettura Harvard prevede l'uso di due memorie separate, una contenente solo dati (memoria dati), l'altra solo le istruzioni (memoria programma), connesse attraverso due bus separati. In questo modo è possibile caricare simultaneamente dati e istruzioni dalle due memorie in un unico ciclo di clock. La maggioranza dei processori dedicati all'elaborazione digitale dei segnali, detti DSP, utilizzano questa architettura.

L'evoluzione dell'architettura Harvard attualmente utilizzata nei DSP prodotti dalla Analog Devices è denominata SHARC (Super Harvard Architecture). Gli elementi che la costituiscono sono:

- Unità di elaborazione: per incrementare drasticamente le prestazioni nel calcolo numerico sono presenti tre unità di elaborazione separate: la ALU (Arithmetic Logic Unit), il moltiplicatore con accumulo (MAC) ed il *barrel shifter*. La ALU si occupa di operazioni aritmetiche (somma, sottrazione, valore assoluto) tra numeri in virgola fissa e in virgola mobile. Il moltiplicatore esegue moltiplicazioni e moltiplicazioni con accumulo in unico ciclo di clock. Il *barrel shifter* esegue manipolazioni elementari su numeri in virgola fissa. Una caratteristica molto utile degli SHARC è la possibilità di accedere al moltiplicatore ed alla ALU simultaneamente. Questo consente di eseguire una serie di operazioni multifunzione in un unico ciclo di clock.
- Unità di generazione degli indirizzi (DAG): Vengono utilizzati due generatori di indirizzi, uno per ciascuna memoria. Questi componenti controllano gli indirizzi inviati alle memorie programma e dati specificando la locazione in cui risiede l'informazione che deve essere letta o scritta.

Essi consentono inoltre l'uso dei cosiddetti buffer circolari [23]. Un buffer circolare è costituito da un insieme di indirizzi che la DAG aggiorna costantemente, scorrendoli in modo circolare. L'idea alla base di queste strutture è che l'ultimo indirizzo dell'insieme è collegato al primo. Come mostrano le fig. 4.2 a) e b), per implementare un

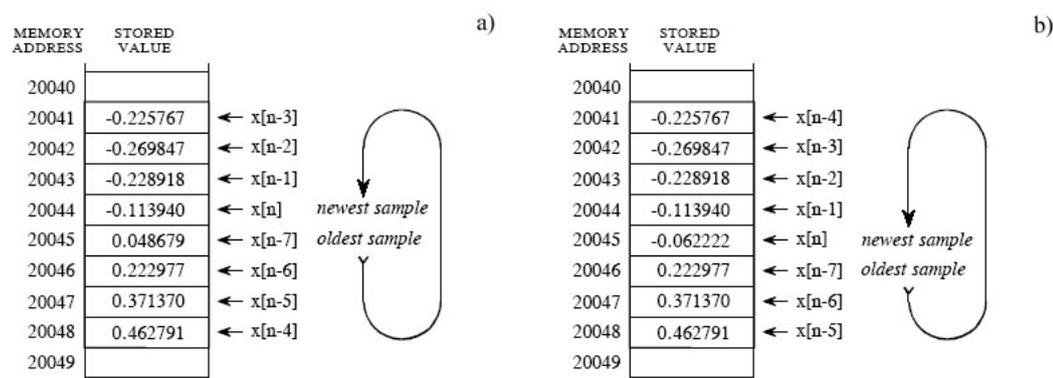


Figura 4.2: Schema di funzionamento un buffer circolare ad 8 campioni: a) mostra il buffer ad un certo istante, b) mostra come appare dopo l'arrivo di un nuovo campione

buffer circolare è necessario un puntatore, ossia una variabile il cui valore è un indirizzo, che indica la locazione del campione più recente. Quando viene acquisito un nuovo campione, esso sostituisce quello più vecchio presente nel buffer ed il puntatore scorre all'indirizzo seguente. In questo modo è necessario modificare un solo valore dell'insieme all'acquisizione di un nuovo campione.

I buffer circolari sono di fondamentale importanza per la nostra applicazione. Infatti, come abbiamo visto nel capitolo precedente, la funzione di trasferimento di un filtro digitale IIR di ordine N può essere tradotta in una corrispondente equazione alle differenze della forma (3.47). Questo algoritmo numerico è ricorsivo nel senso che è possibile ricavare lo stato interno, e da questo l'uscita del filtro, partendo dall'ingresso x_n , dai coefficienti del filtro e dagli stati interni w_{n-1}, \dots, w_{n-N} corrispondenti agli istanti precedenti. Questi ultimi devono essere mantenuti in memoria e continuamente aggiornati all'acquisizione di nuovi campioni, operazione che può essere eseguita da un buffer circolare in modo molto efficiente.

- Memoria: si utilizza un'architettura Harvard ibrida. Infatti sebbene la memoria sia divisa in due blocchi indipendenti connessi al processore da bus separati, la memoria dati può contenere solo dati mentre la memoria programma può contenere sia istruzioni che dati. In questo modo è possibile caricare due operandi in un ciclo di clock a condizione che l'istruzione venga fornita dalla memoria cache.

Come vedremo nella prossima sezione, oltre a questi componenti sono presenti nei DSP SHARC anche un sequenziatore di programma con numerose

funzioni dedicate alla gestione delle istruzioni non sequenziali, una memoria cache delle istruzioni ad alta velocità ed un controllore I/O connesso alla memoria dati.

L'architettura SHARC risulta quindi adeguatamente ottimizzata per i nostri scopi. Come vedremo nella prossima sezione, il processore scelto per il nostro progetto appartiene a questa famiglia di DSP.

4.3 L'ADSP21160N

L'ADSP21160N è un DSP a virgola mobile a 32 bit, con frequenza di clock di 95 MHz, appartenente alla 3^a generazione della famiglia SHARC [24]. Esso consente di effettuare un massimo di 570 MFLOPS (MFLOPS=milioni di operazioni in virgola mobile/secondo). Come vedremo nel prossimo capitolo la scheda elettronica oggetto di questa tesi sarà basata su un sistema multiprocessore costituito da 6 ADSP21160N.

Nella seguente tabella sono mostrati i tempi di calcolo t_{CPU} ottenibili con questo DSP negli algoritmi tipici dell'elaborazione digitale del segnale

Algoritmo	t_{CPU}
FFT complessa a 1024 punti	96 μ s
Filtro FIR (per tap)	5.25 ns
Filtro IIR del II ordine	21 ns
Prodotto matriciale	
$[3 \times 3] \times [3 \times 1]$	47.25 ns
$[4 \times 4] \times [4 \times 1]$	84 ns
Divisione (y/x)	31.5 ns

Il diagramma a blocchi del processore è mostrato in fig. 4.3 Nelle prossime sezioni verranno analizzate in dettaglio le caratteristiche chiave della sua architettura interna.

4.3.1 Bus interni

Nell'ADSP21160N sono presenti sei bus interni:

- PM, PMD: i bus degli indirizzi e dei dati della memoria programma
- DMA, DMD: i bus degli indirizzi e dei dati della memoria dati. In particolare il DMD consente di trasferire il contenuto di ogni registro del processore o qualsiasi locazione di memoria in un singolo ciclo di clock
- IOA, IOD: i bus degli indirizzi e dei dati che consentono di accedere alla memoria interna del processore senza ritardare l'esecuzione delle istruzioni

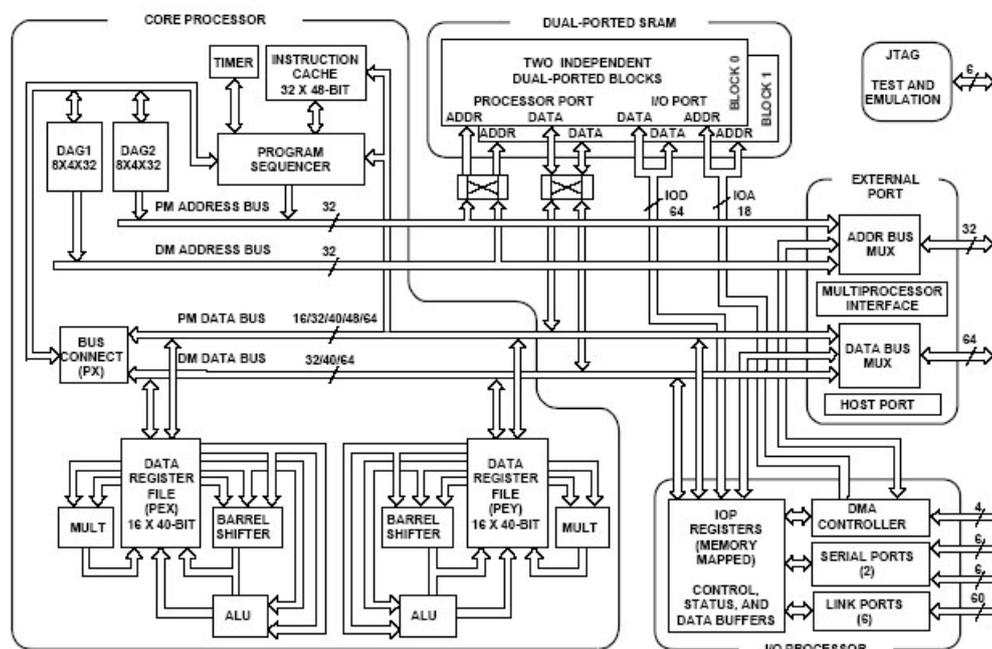


Figura 4.3: Schema a blocchi del processore ADSP21160N.

PMA, DMA ed IOA sono tutti larghi 32 bit mentre PMD, DMD, IOP sono larghi 64 bit. Nell'ADSP21160N il principio base dell'architettura Harvard è leggermente violato. Infatti mentre la memoria dati contiene solo operandi, la memoria programma può contenere sia istruzioni che dati. Questa caratteristica consente al processore l'accesso contemporaneo a due dati in un singolo ciclo di clock.

4.3.2 Unità di elaborazione

Nel processore sono presenti due unità di elaborazione identiche, dette PEX e PEY. Ciascuna di esse contiene un registro dati multiplo di 16 x 40 bit e tre unità di calcolo: una ALU, un moltiplicatore con accumulo ed un *barrel shifter*. Le unità di calcolo, come mostra la fig.4.3, sono disposte in parallelo in modo che l'uscita di una qualsiasi unità possa essere usata come ingresso in un'altra. Questo tipo di architettura è detta non *pipelined* in quanto, a differenza dei processori tradizionali i dati in ingresso o in uscita dalle unità devono attraversare il registro dati multiplo. Costituito da 32 registri a 40 bit, 16 primari e 16 secondari, consente, grazie a due porte ad alta velocità, il trasferimento dati tra i bus PM e DM e le unità di calcolo nonché la memorizzazione locale degli operandi e dei risultati delle operazioni. Le tre unità di calcolo gestiscono differenti tipi di operazioni:

ALU Esegue operazioni aritmetiche su numeri a 32 bit a virgola fissa in complemento a due, numeri a virgola mobile in precisione singola secondo lo standard IEEE754-1987, numeri a virgola mobile in precisione estesa con 31 bit di mantissa ed 8 bit di esponente. Le istruzioni eseguibili dalla ALU includono:

- addizioni, sottrazioni, medie con numeri in virgola fissa o mobile
- manipolazione di numeri in virgola mobile
- AND, OR, XOR, NOT logici con numeri in virgola fissa
- conversioni tra formati numerici.

Il DSP trasferisce nel registro multiplo gli operandi in ingresso alla ALU nella prima metà di un ciclo di clock, quindi i risultati dell'operazione nella seconda metà. In questo modo la lettura degli operandi, l'esecuzione dell'operazione e la scrittura dei dati nel registro multiplo avvengono in unico ciclo di clock.

Moltiplicatore con accumulo Il moltiplicatore esegue moltiplicazioni e moltiplicazioni con accumulo sia con numeri a 32 bit in virgola fissa che con numeri in virgola mobile in precisione singola ed estesa. Mentre le operazioni in virgola mobile eseguite dal moltiplicatore hanno operandi e risultati a 32 o 40 bit, le operazioni in virgola fissa hanno operandi a 32 bit ma risultati ad 80 bit. Analogamente alla ALU la lettura degli operandi dal registro multiplo, l'esecuzione delle operazioni e la scrittura dei risultati avvengono in un unico ciclo di clock.

Come abbiamo visto nel capitolo precedente, la possibilità di eseguire una moltiplicazione con accumulo in un singolo ciclo di clock è di fondamentale importanza nell'implementazione di filtri digitali IIR o FIR e rappresenta una delle caratteristiche peculiari dei DSP. L'ADSP21160N è capace di eseguire un massimo di 200 MMAC (MMAC = milioni di moltiplicazioni con accumulo/secondo).

Barrel shifter Il *Barrel shifter* esegue vari tipi di manipolazioni su operandi in virgola fissa a 32 bit.

Usando la già citata architettura parallela delle tre unità di calcolo è possibile eseguire operazioni multifunzione. Queste istruzioni vengono completate in un unico ciclo di clock e sono costituite da due operazioni che vengono effettuate contemporaneamente dal moltiplicatore e dalla ALU su operandi diversi. Si possono quindi eseguire moltiplicazioni o moltiplicazioni con accumulo contemporaneamente a somme, sottrazioni o qualsiasi altra operazione della ALU. Si possono anche effettuare due operazioni della ALU nello stesso tempo sempre in un singolo ciclo di clock.

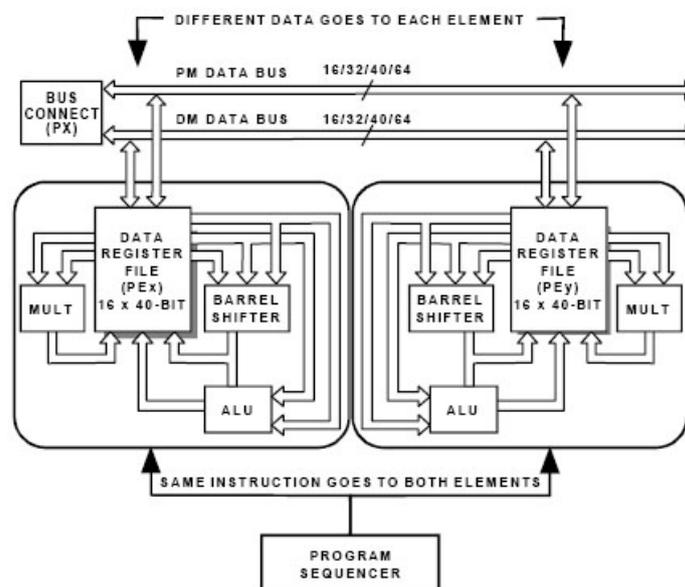


Figura 4.4: Schema di funzionamento della modalità SIMD supportata dall'ADSP21160N

La presenza di una doppia unità di elaborazione, PEX e PEY, consente infine di implementare la cosiddetta modalità SIMD. L'ADSP21160N è il primo processore della famiglia SHARC a possedere questa caratteristica. Come indica l'acronimo (Singola Istruzione Molteplici Dati), in questa modalità il DSP esegue le seguenti operazioni in successione all'interno di un singolo ciclo di clock (vedi fig. 4.5):

- il sequenziatore di programma (vedi la prossima sezione) invia una singola istruzione alle unità di calcolo di entrambe le unità di elaborazione
- vengono caricati due insiemi di dati dalla memoria, uno per ciascuna unità di elaborazione
- la stessa istruzione viene eseguita contemporaneamente in entrambe le unità di elaborazione
- i risultati delle operazioni vengono trasferiti alla memoria

Molti algoritmi tipici dell'elaborazione dei segnali, tra cui l'implementazione di filtri IIR del II ordine, implementati in modalità SIMD, possono essere eseguiti in metà tempo rispetto al modo standard SISD (Singola Istruzione Singolo Dato).

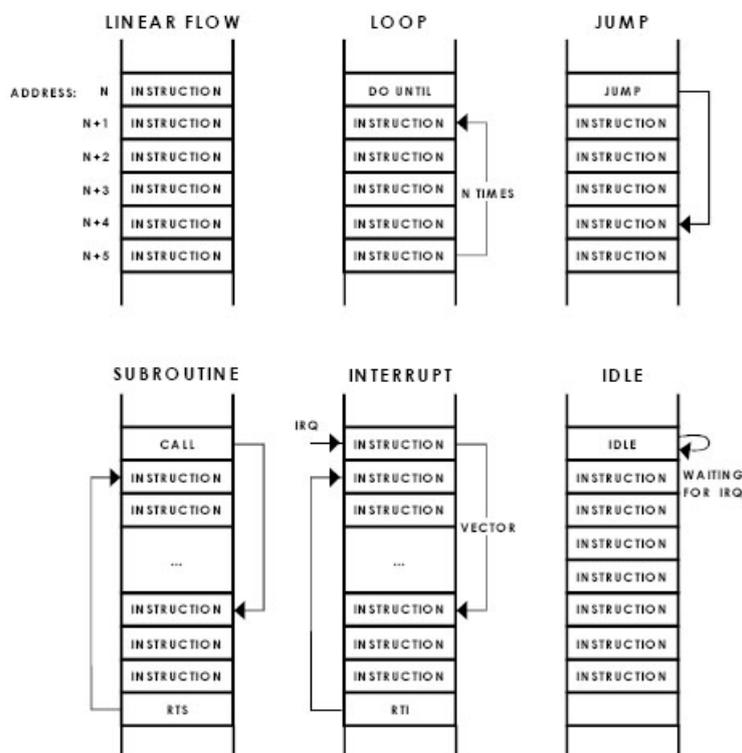


Figura 4.5: Strutture non sequenziali

4.3.3 Sequenziatore di programma

Il sequenziatore di programma si occupa del corretto flusso di istruzioni, fornendo costantemente ai sottosistemi del processore l'indirizzo dell'istruzione successiva da eseguire. Sebbene nella maggior parte dei casi il flusso sia lineare e le istruzioni vengano eseguite in sequenza, può accadere che il programma preveda delle strutture non sequenziali. I principali tipi di queste strutture sono mostrate in fig. 4.5. Esse includono

- *Loop*: lo stesso insieme di istruzioni viene eseguito n volte
- *Subroutine*: il processore interrompe il flusso sequenziale per eseguire istruzioni in un'altra sezione della memoria programma
- Salto: il flusso delle istruzioni viene trasferito in modo permanente in un'altra sezione della memoria programma
- Interruzione: *subroutine* in cui un evento induce l'esecuzione di una *routine*
- *IDLE*: istruzione che blocca l'elaborazione in attesa di un'interruzione.

L'esecuzione sequenziale delle istruzioni, utilizzata anche dei processori tradizionali, detta anche *pipelined*, prevede l'uso di tre cicli di clock:

- ciclo di caricamento o *fetch*: il DSP legge le istruzioni dalla memoria programma o dalla memoria *cache*.
- ciclo di decodifica: il DSP decodifica le istruzioni generando le condizioni che controllano l'esecuzione delle istruzioni stesse.
- ciclo di esecuzione: il DSP esegue le operazioni contenute nell'istruzione in un singolo ciclo di clock.

Come mostra la seguente tabella, sebbene questi cicli si sovrappongano, dopo un ritardo iniziale di due cicli di clock, il processore esegue un'istruzione per ciclo:

Ciclo	Caricamento	Decodifica	Esecuzione
1	A		
2	B	A	
3	C	B	A
4	D	C	B
5	E	D	C

Un registro detto contatore di programma (PC) contiene gli indirizzi a 24 bit di tutte le istruzioni che sono correntemente nella fase di caricamento, decodifica o esecuzione.

La gestione delle istruzioni non sequenziali precedentemente elencate avviene attraverso il cosiddetto *branching*. Esso si verifica quando un'istruzione viene eseguita ad una locazione diversa da quella dell'indirizzo successivo. Vi sono una serie di parametri che possono essere definiti per il *branching*: si distinguono infatti

- *Branch* immediati: il processore svuota la *pipeline* e la riempie con le istruzioni previste dal *branch*. L'esecuzione di un *branch* immediato è molto sfavorevole in termini di tempo di elaborazione in quanto comporta la perdita di due cicli di clock. Questo è mostrato nella seguente tabella dove NOP (No Operation) indica uno stato di inattività e j indica la prima istruzione del *branch*:

Ciclo	Caricamento	Decodifica	Esecuzione
1	$n + 2$	$n + 1 \Rightarrow \text{NOP}$	n
2	j	$n + 2 \Rightarrow \text{NOP}$	NOP
3	$j + 1$	j	NOP
4	$j + 2$	$j + 1$	j

- *Branch* ritardati: utilizzando i *branch* ritardati è possibile eseguire istruzioni non sequenziali senza perdere alcun ciclo di clock. Infatti,

come è mostrato nella seguente tabella, il DSP esegue le due istruzioni successive al *branch* mentre la prima istruzione non sequenziale è in fase di caricamento e di decodifica.

Ciclo	Caricamento	Decodifica	Esecuzione
1	$n + 2$	$n + 1$	n
2	j	$n + 2$	$n + 1$
3	$j + 1$	j	$n + 2$
4	$j + 2$	$j + 1$	j

Nel caso il *branch* preveda un ritorno, come avviene ad esempio nelle *subroutine*, questo viene effettuato al ciclo $n + 3$ e non all' $n + 1$, guadagnando quindi due cicli di clock sul *branching* immediato.

Queste caratteristiche del sequenziatore di programma dell'ADSP21160N rendono possibile una notevole ottimizzazione del codice ed una piena prevedibilità dei tempi di calcolo, caratteristiche fondamentali di un processore dedicato ad applicazioni in tempo reale.

4.3.4 DAG1 e DAG2

Nell'ADSP21160N sono disponibili due generatori di indirizzi, DAG1 e DAG2, che si occupano di fornire i corretti indirizzi di memoria quando vengono trasferiti dati la memorie ed i registri interni. La presenza di due DAG consente al processore, in caso di lettura o scrittura di due operandi, di generare i relativi indirizzi simultaneamente. DAG1 fornisce gli indirizzi della memoria programma mentre DAG2 genera gli indirizzi della memoria dati. Ciascuno dei due generatori supporta l'usa dei già citati buffer circolari.

4.3.5 Memoria

L'ADSP21160N contiene 4 Mbit di memoria statica, divisa in due blocchi da 2 Mbit, che può contenere codice e dati in differenti combinazioni. Questa memoria è di tipo Dual-Port nel senso che è possibile accedere a ciascuno dei due blocchi indipendentemente e in unico ciclo di clock. Come abbiamo già detto, al fine di poter eseguire due trasferimenti dati in un ciclo di clock, la netta divisione tra memoria programma e memoria dati, caratteristica dell'architettura Harvard, non è del tutto rispettata. La configurazione più efficiente è quella in cui un blocco memorizza i dati, usando il bus DMD per i trasferimenti, mentre l'altro blocco memorizza sia indirizzi che dati, usando il bus PMD per i trasferimenti. La memoria può essere suddivisa in parole da 16,32,48 o 64 bit.

Il bus dedicato al trasferimento esterno degli indirizzi è detto ADDR0-31 mentre quello usato per il trasferimento esterno dei dati e delle istruzioni

è detto DATA0-63. Quest'ultimo è largo 64 bit in modo da poter trasferire numeri in virgola mobile a 40 bit ed istruzioni di 48 bit.

L'ADSP21160N consente inoltre di aggiungere fino a 4 Gword di memoria esterna statica o dinamica. Lo spazio degli indirizzi è unico ed include la memoria esterna e lo spazio di memoria di tutti i DSP usati in un'eventuale configurazione multiprocessore. E' possibile aggiungere memoria esterna sia statica (SRAM) che dinamica (DRAM). Nel caso della SRAM, lo spazio di memoria esterna viene diviso in quattro banchi della stessa capacità che possono essere selezionati dal DSP attraverso i segnali $\overline{MS0} - \overline{3}$. Nel caso della DRAM invece lo spazio di memoria è mappato in un unico banco, il banco 0, ed è necessario l'uso di un controllore ossia di un circuito esterno al processore che consenta di gestire la memoria. E' infine possibile aggiungere una memoria programmabile o EPROM da usare per la memorizzazione del codice e dei dati necessari all'inizializzazione del processore. Come vedremo nel cap. 5, il nostro progetto prevede l'uso di tutti e tre i tipi di memoria che possono essere aggiunti al DSP.

4.3.6 *Link ports e configurazioni multiprocessore*

Nell'ADSP21160N sono presenti sei porte di connessione o link-port. Queste porte bidirezionali, costituite da 8 linee dati, una linea di clock ed un *acknowledge*, consentono di connettere il processore ad altri DSP o a periferiche esterne. Le link port possono operare alla stessa frequenza di clock del DSP, 95 MHz, e permettono di trasferire fino ad 8 bit di dati per ciclo: il flusso dati massimo ottenibile da una link-port è quindi di 95 MB/s.

Caratteristica fondamentale delle link-port è che possono operare simultaneamente ed in modo indipendente. Attraverso le porte di connessione è possibile realizzare numerose configurazioni multiprocessore riassunte nella fig. 4.6.

Esse si dividono essenzialmente in due tipi di architetture:

- Architettura *data-flow*: i DSP sono connessi in successione dalle link-port. Questa configurazione viene usata in applicazioni in cui è richiesto un elevato flusso dati ma non la capacità di eseguire più algoritmi contemporaneamente. Il codice deve essere infatti sviluppato in modo da suddividere l'algoritmo sequenzialmente tra i processori e da sfruttare i trasferimenti in linea dei dati.
- Architettura *cluster*: i DSP sono connessi ad un bus parallelo che consente l'accesso alle memorie interne dei singoli processori e ad un'eventuale memoria esterna condivisa. In questa configurazione possono essere connessi fino a 6 DSP. Ciascuno di essi è identificato da tre segnali fissati, ID0-3, che rappresentano il processore con un numero binario da 001 a 110.

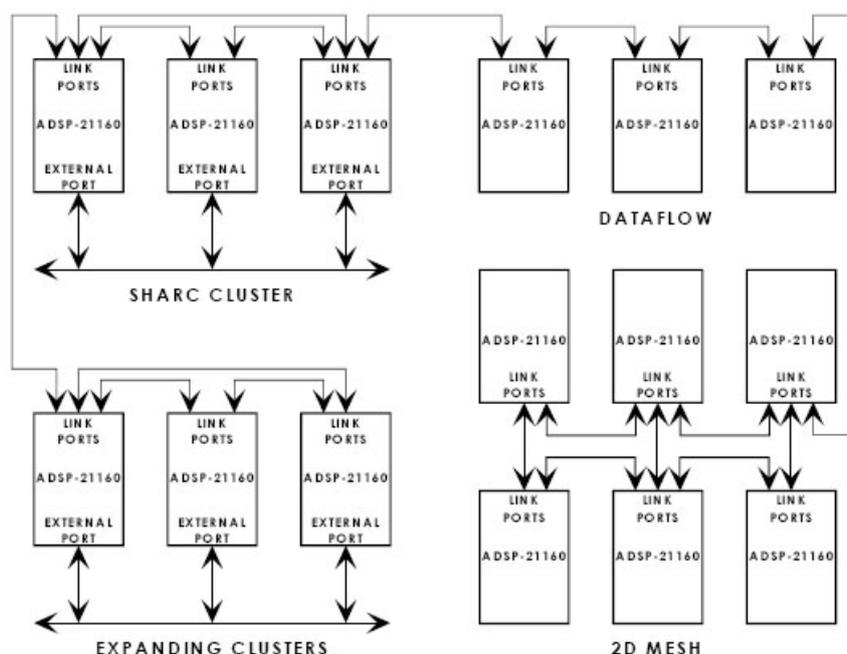


Figura 4.6: Alcune possibili configurazioni multiprocessore dell'ADSP21160N

Il bus può essere controllato da ognuno dei DSP singolarmente che ne diventa in questo modo il *master*. Il *master* del bus ha accesso sia alla memoria esterna che alla memoria e parte dei registri interni di tutti i processori del sistema. Vi sono due possibili criteri, detti arbitraggi, con cui è possibile assegnare il controllo del bus in caso di richieste in conflitto: la priorità fissa, in cui viene favorito il processore con ID più basso, e la priorità a rotazione, in cui ad ogni DSP viene assegnata in rotazione la priorità più alta.

L'architettura *cluster* viene usata in quelle applicazioni in cui è necessario combinare l'elevato flusso dati con la possibilità di eseguire contemporaneamente più algoritmi. Come mostra la fig. 4.6, sono disponibili varie tipologie alternative per questa configurazione. In particolare, disponendo i processori in una matrice bidimensionale o tridimensionale (*2/3D mesh*), è possibile usare la già citata modalità SIMD. Usando l'ADSP21160N questa modalità può essere dunque implementata a due livelli: nel singolo processore, grazie alle unità PEx e PEy, ed in un sistema multiprocessore, connettendo i DSP in una matrice $n \times n$. Come vedremo nel prossimo capitolo, quattro dei sei processori usati nel nostro progetto sono connessi in configurazione *2D mesh*.

4.3.7 Porte seriali

L'ADSP21160N è infine fornito di due porte seriali sincrone, SPORT0 e SPORT1, che consentono di interfacciarlo con numerosi tipi di periferiche. Queste porte possono funzionare a metà della frequenza di clock del processore consentendo un flusso dati massimo di 47.5 Mbit/s.

Capitolo 5

Progetto del circuito

5.1 Schema generale del circuito

Come è emerso dal cap. 3 la potenza di calcolo della scheda DSP utilizzata nell'esperimento non consente di incrementare le prestazioni del controllo locale rispetto al loro livello attuale. Questo si traduce nell'impossibilità di assicurare un corretto funzionamento dell'interferometro nelle varie condizioni sperimentali. Allo stesso tempo il DSP96002, l'unità di elaborazione utilizzata dalla scheda, non viene più prodotta e non è quindi più possibile effettuare la manutenzione delle schede DSP attualmente in uso nell'esperimento. Emerge quindi la necessità di sviluppare un nuovo circuito dedicato al controllo delle sospensioni di Virgo.

Tale sviluppo si inserisce in un ampio progetto di rinnovamento di tutta l'elettronica del controllo locale che coinvolge anche il già citato laboratorio LAPP di Annecy. Quest'ultimo si occuperà di nuove DOL e nuove schede di Timing. Lo schema a blocchi generale del nuovo sistema di controllo locale è mostrato in fig. 5.1. Esso include

- la scheda DSP
- la DOL
- la scheda di Timing
- uno o più cpu modulari definite dallo standard PC/104.
- le schede ADC e DAC.

I primi quattro circuiti elettronici sopra elencati saranno riuniti in un'unica scheda madre in formato VME. I due lati della scheda, con le dimensioni dei vari sottosistemi, sono mostrati in fig. 5.2. Il nostro circuito sarà collegato alla scheda madre attraverso dei connettori compatibili con lo standard PMC (PCI Mezzanine Card).

All'interno dell'appendice A sono riuniti gran parte degli schematici elettrici della nuova scheda DSP. Essi sono stati realizzati facendo uso del software CAD (Computer Aided Design) Capture CIS prodotto dalla Cadence. In progetti come questo che l'uso di un numero molto elevato di componenti e di segnali, gli schematici devono essere organizzati secondo una struttura gerarchica. Si definisce infatti uno schematico principale, detto modulo root (nel nostro caso lo schema 1), che rappresenta il progetto completo ossia il livello più alto della gerarchia. Il modulo root può contenere sia componenti singoli che blocchi gerarchici. Ciascuno blocco gerarchico contiene a sua volta uno più schematici che sono connessi al resto del progetto attraverso delle porte, dette porte gerarchiche. In questo modo è possibile suddividere un progetto complesso in una serie di blocchi semplici. Il controllo degli errori avviene attraverso un tool denominato DRC (Design Rule Check). Esso, in base alle proprietà assegnate alle porte gerarchiche e ai contatti dei singoli componenti, verifica la correttezza delle varie connessioni.

A partire dal prossimo paragrafo vengono descritti i vari componenti della scheda.

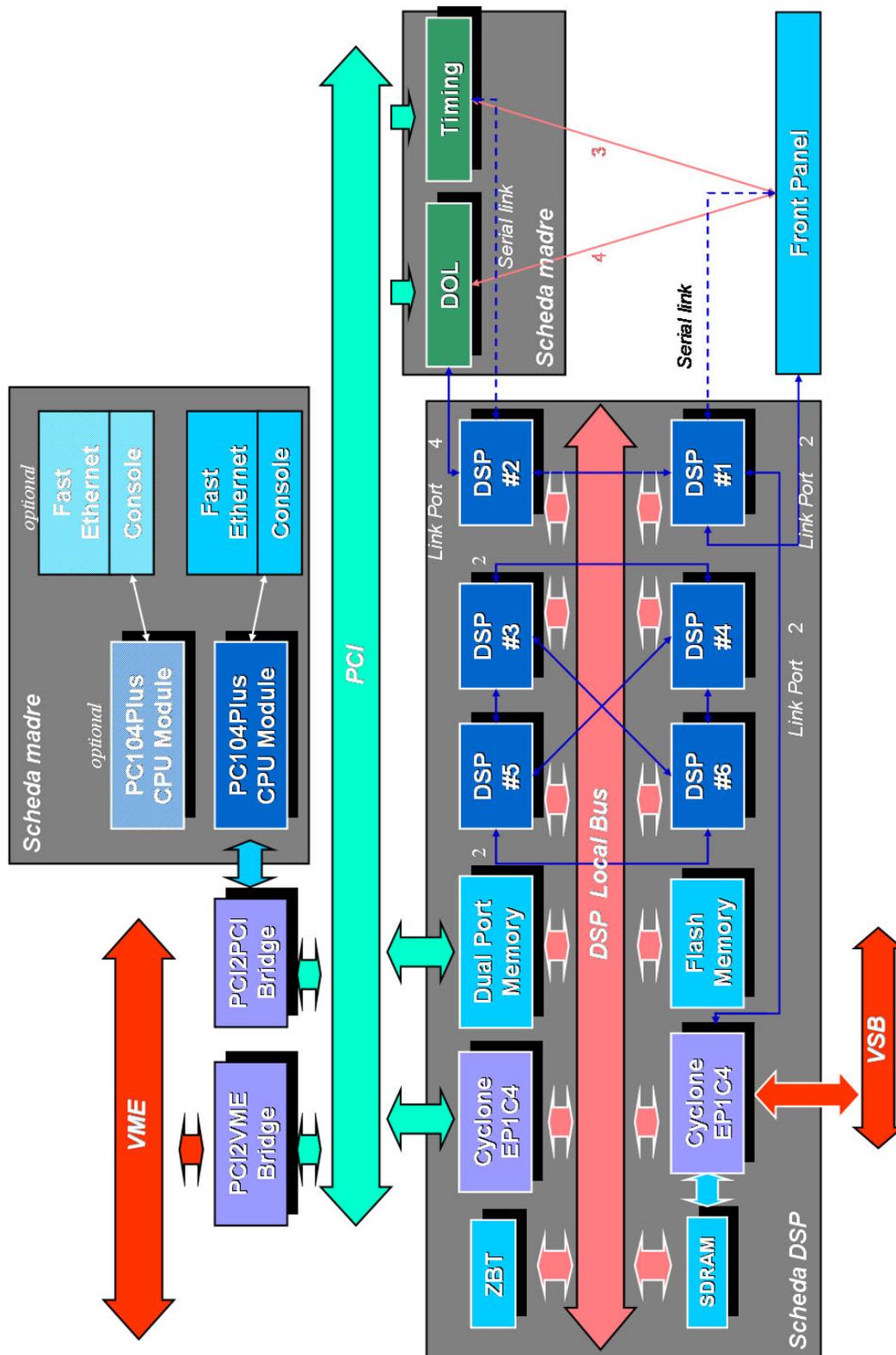


Figura 5.1: Schema a blocchi del nuovo controllo locale di Virgo



Figura 5.2: Scheda madre in formato VME che ospiterà gran parte dei componenti del controllo locale

5.1.1 Unità di elaborazione

Per il nostro progetto, come già detto in precedenza è stato scelto un sistema multiprocessore con architettura *cluster* costituito da sei ADSP21160N. La topologia del sistema, mostrata sia nello schema a blocchi di fig. 5.1 che nello schema elettrico 1, è di tipo ibrido. Infatti, sebbene tutti i DSP condividano un unico bus locale (DSP Local Bus), due processore, DSP1 e DSP2, sono connessi tra loro da una coppia di link-port mentre gli altri quattro sono disposti in una matrice 2x2. I processori all'interno del circuito hanno infatti funzioni differenti:

- DSP con ID=001 (DSP1): verrà utilizzato esclusivamente per il controllo in tempo reale delle sospensioni, compito attualmente svolto dal DSP96002. Questo prevede, come abbiamo visto nel cap. 3, l'implementazione di filtri digitali in cascata e lo scambio di dati, attraverso il bus VSB, con i convertitori ADC e DAC connessi ai sensori e agli attuatori presenti sul SA. Due link-port sono connesse al DSP2 in modo da poter trasferire dati con la DOL e la scheda di Timing e altre due sono collegate all'FPGA EP1C4B che consente l'interfacciamento col bus VSB. Rimangono disponibili due link-port che vengono inviate al pannello frontale insieme ad una delle due porte seriali di cui è fornito il processore.

Grazie alla potenza di calcolo dell'ADSP21160N, mantenendo l'attuale frequenza di campionamento $f_c = 10$ KHz sarà ora possibile negli $80 \mu s$ disponibili per l'elaborazione (v. cap. 3), eseguire calcoli (implementazione di filtri, prodotti matriciali) equivalenti ad un totale di quasi 4000 poli e zeri. Alternativamente potrà essere incrementata la frequenza di campionamento in modo da aumentare il margine di fase M_F alla frequenza di guadagno unitario. Si potrà così estendere la banda del controllo locale fino a Hz.

- DSP con ID=010 (DSP2): lo scambio dati tra il circuito e le nuove schede DOL e Timing, compito attualmente svolto dal DSP96002 attraverso la CPLD1, viene affidato al DSP2. Quest'ultimo riceve i dati del DSP1 attraverso due link-port e, dopo averli filtrati, li invia alla DOL tramite 4 link-port e alla scheda di Timing usando una porta seriale. Quindi il DSP2 può scambiare dati con la DOL, con una banda di 380 MB/s, e con la scheda di Timing, con una banda di 47.5 Mbit/s.
- DSP con ID=011-110 (DSP3-6): questi quattro processori sono disponibili per applicazioni future quali l'analisi dei dati dell'esperimento o lo sviluppo di altre forme di controllo. Sono stati connessi seguendo la topologia 2D *mesh*, descritta nel cap. 4, in modo da poter sfruttare

la modalità SIMD nello sviluppo degli algoritmi. Utilizzando la configurazione di fig. 4.6 con quattro processori, due link-port per ogni DSP sarebbero rimaste inutilizzate: è stato quindi scelto di collegare il DSP4 con il DSP5 ed il DSP3 con il DSP6. In questo modo, usando le link-port, ciascun processore può accedere in modo diretto agli altri tre.

5.1.2 Le FPGA Cyclone EP1C4

Come si può vedere dallo schema elettrico 1, le funzioni delle 2 FPGA sono molteplici.

EP1C4A E' dedicata alle seguenti funzioni:

- interfaccia il bus locale dei DSP con il bus PCI. La presenza di questo è di fondamentale importanza per poter interfacciare il circuito con una qualsiasi CPU e con il bus VME.

Il PCI è uno standard nato nel 1992 con la necessità di fornire un'interfaccia ad alta velocità tra i processori della famiglia x86 e le periferiche di visualizzazione grafica (ad esempio le schede video) [25]. A partire da una delle sue ultime revisioni, la 2.2 del 1998, il bus PCI può funzionare con dati sia di 32 che di 64 bit e con frequenze di clock di 33 o 66 MHz: il flusso dati massimo ottenibile varia quindi da 132 MB/s a 528 MB/s. A seconda della configurazione scelta il bus PCI comprende da un minimo di 47 segnali ad un massimo di 98. E' possibile accedere al bus con latenze molto ridotte: ad esempio nel caso di un ciclo di scrittura effettuato a 33 MHz la latenza vale 60 ns.

Caratteristica fondamentale del bus PCI è il multiplexing degli indirizzi: al fine di ridurre al minimo le dimensioni dei connettori, vengono utilizzati gli stessi contatti per scambiare dati ed indirizzi. Un trasferimento dati consiste quindi in una fase di invio dell'indirizzo ed in una o più fasi di invio dati.

Nel nostro progetto per poter interfacciare tra loro i bus PCI e VME è necessario un bridge, ossia un componente progettato per rendere compatibili due bus definiti da standard differenti. A tal fine sarà installato sulla scheda madre il chip Universe II prodotto dalla Tundra. Esso è pienamente compatibile con lo standard PCI a 33 MHz sia nella sua versione a 32 bit che in quella a 64 bit.

- consente il trasferimento veloce di dati tra il bus locale ed il bus PCI attraverso la memoria Dual-Port prevista nel circuito.

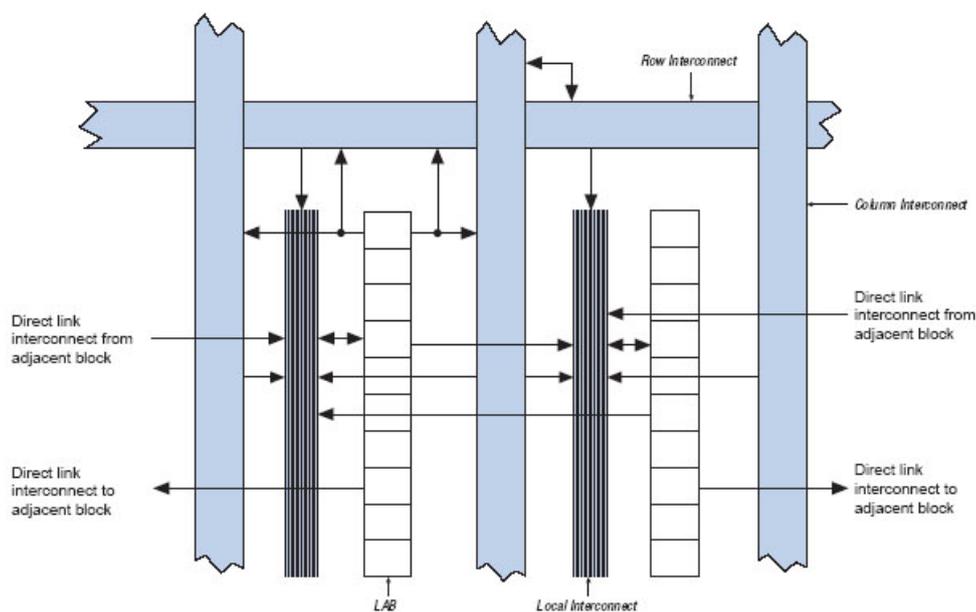


Figura 5.3: *Struttura interna di una FPGA Cyclone.*

- inizializza il sistema multiprocessore usando il codice contenuto nella memoria Flash.

EP1C4B E' dedicata alle seguenti funzioni:

- consente il controllo del banco di memoria esterna SDRAM da parte dei DSP.
- interfaccia il bus locale dei DSP con il bus VSB usando due link-port del DSP1.

Le FPGA (Field Programmable Gate Array) costituiscono l'evoluzione delle logiche programmabili CPLD usate nell'attuale sistema di controllo. Questi dispositivi vengono generalmente usati con lo scopo di integrare in unico componente circuiti logici anche molto complessi. Le FPGA scelte per il nostro progetto appartengono alla famiglia delle logiche Cyclone prodotte dalla Altera [26].

Come mostra la fig. 5.3, questi componenti hanno un'architettura interna costituita da una matrice bidimensionale di elementi detti Logic Array Block (LAB). Ciascuno di essi contiene dieci elementi logici (LE). L'elemento logico è la più piccola unità dell'architettura delle FPGA capace di implementare una qualsiasi funzione logica. I trasferimenti di segnali tra

elementi logici appartenenti allo stesso LAB avvengono attraverso delle connessioni dette local interconnect. Analogamente, tra le righe e le colonne della matrice di LAB, sono presenti connessioni dette rispettivamente row interconnect e column interconnect. Il numero di elementi logici fornisce una misura del livello di complessità del circuito logico implementabile nella FPGA. La Cyclone EP1C4 possiede una matrice costituita da 13x24 LAB per un totale di 4120 elementi logici.

La grande versatilità delle FPGA risiede nella possibilità di essere riprogrammate a piacere. In questo modo parte della progettazione di un circuito si traduce nello sviluppo di codice adeguato a descriverne le caratteristiche logiche. I linguaggi di programmazione che consentono di descrivere le caratteristiche di un circuito elettronico sono detti HDL (Hardware Description Languages). A differenza dei linguaggi tradizionali, come il C o il FORTRAN, essi posseggono una sintassi e dei comandi che permettono di esprimere l'evoluzione temporale di un singolo processo o di più processi che operano contemporaneamente. I linguaggi HDL più usati sono il VHDL ed il VERILOG.

Il codice necessario ad implementare le funzioni descritte all'inizio del paragrafo deve essere ancora sviluppato. Il linguaggio HDL utilizzato sarà molto probabilmente il VERILOG. Per l'interfaccia con il bus PCI verrà usata la macro PCICORE prodotta dalla Altera.

5.1.3 Memoria

SDRAM Nel progetto sono previsti 8 banchi di memoria SDRAM da 512 Mbit ciascuno per un totale di 512 MB disponibili. Il componente usato è l'MT48LC64M8A2 prodotto dalla Micron.

In questi dispositivi ciascun bit di dati è memorizzato in un singolo condensatore e la quantità di carica elettrica immagazzinata determina il livello del segnale. In un chip di memoria i condensatori sono disposti in una matrice bidimensionale. A causa di diversi tipi di perdite, la carica, e quindi il bit memorizzato, scompare in meno di un secondo. E' dunque necessario aggiungere della logica esterna che consenta di riscrivere periodicamente il dato.

Una delle caratteristiche principali delle DRAM è il multiplexing degli indirizzi. In questa tecnica l'indirizzo di un dato è suddiviso in due gruppi che vengono trasferiti in successione su gli stessi contatti del componente. Ad esempio per leggere una locazione nella matrice di condensatori, il circuito di controllo dapprima calcola il numero di riga corrispondente e lo trasmette attraverso i contatti di indirizzo del chip. Quindi asserisce il segnale RAS (Row Array Select) che avvia la lettura dei bit contenuti nella riga. Internamente la DRAM connette la riga selezionata di condensatori ad un banco di amplificatori che ne

determina la carica accumulata. A questo punto il circuito di controllo calcola la colonna corrispondente e asserisce il segnale CAS (Column Array Select) che avvia la lettura dei bit contenuti nella colonna. Dopo un ritardo, detto tempo di accesso CAS, i dati vengono infine trasferiti ai contatti di I/O del chip. Tra un'operazione di lettura e scrittura è necessario un intervallo di tempo, detto intervallo di precarica, all'interno del quale i segnali di RAS e CAS sono riportati al loro livello originario. Nel caso dei moduli utilizzati nel nostro circuito il tempo di accesso CAS è di 6 ns mentre l'intervallo di precarica vale.

ZBT Oltre al banco di memoria SDRAM, sulla nuova scheda DSP sono disponibili 2 moduli da 512K x 36 bit per consentire l'accesso veloce a ridotte quantità di dati di particolare rilevanza. Il modello utilizzato è l'IDT71T75702 prodotto dalla IDT. A differenza delle memorie SDRAM, le memorie statiche non richiedono la riscrittura periodica dei dati e possono essere gestite dall'ADSP21160N senza alcun controllore. Inoltre nelle memorie ZBT (Zero Bus Turnaround), non ci sono cicli di clock persi tra un'operazione di lettura ed una di scrittura o viceversa: se gli indirizzi ed i segnali di controllo giungono in ingresso alla ZBT in un certo ciclo di clock, i dati selezionati sono già disponibili nel ciclo successivo. Queste caratteristiche rendono il tempo di accesso alla memoria molto ridotto (7.5 ns nel caso della IDT71T75702).

Dual-Port Al fine di ottenere trasferimenti dati ad alta velocità tra il bus locale ed il bus PCI senza interferire con l'elaborazione del DSP1, nel nostro circuito viene utilizzata la memoria Dual-Port IDT70T3519 prodotta dalla IDT. Essa ha una capacità di 256K x 36 bit e può operare fino a frequenze di clock di 200 MHz con un tempo di accesso ai dati di 3.4 ns.

Le memorie Dual-Port sono memorie statiche che consentono l'accesso simultaneo alla stessa locazione attraverso due porte ad alta velocità. Come si può notare dallo schema a blocchi della IDT70T3519, mostrato in fig. 5.4, le celle di memoria e la logica interna delle Dual-Port sono identiche per entrambi i lati della memoria garantendo così la possibilità di accedere simultaneamente ad ogni indirizzo da parte delle due porte.

FLASH Al fine di memorizzare il codice ed i dati necessari all'inizializzazione dei 6 DSP, nel nostro progetto viene usata una memoria FLASH da 512 x 8 bit: la M29W400D prodotta dalla ST Microelectronics. Le FLASH sono memorie non volatili che immagazzinano dati usando una matrice di transistor, detti celle, ciascuno dei quali memorizza un singolo bit d'informazione. La M29W400D è divisa in blocchi da 8, 16 e 64 KB. Ciascun blocco può essere cancellato indipendentemente dagli altri.

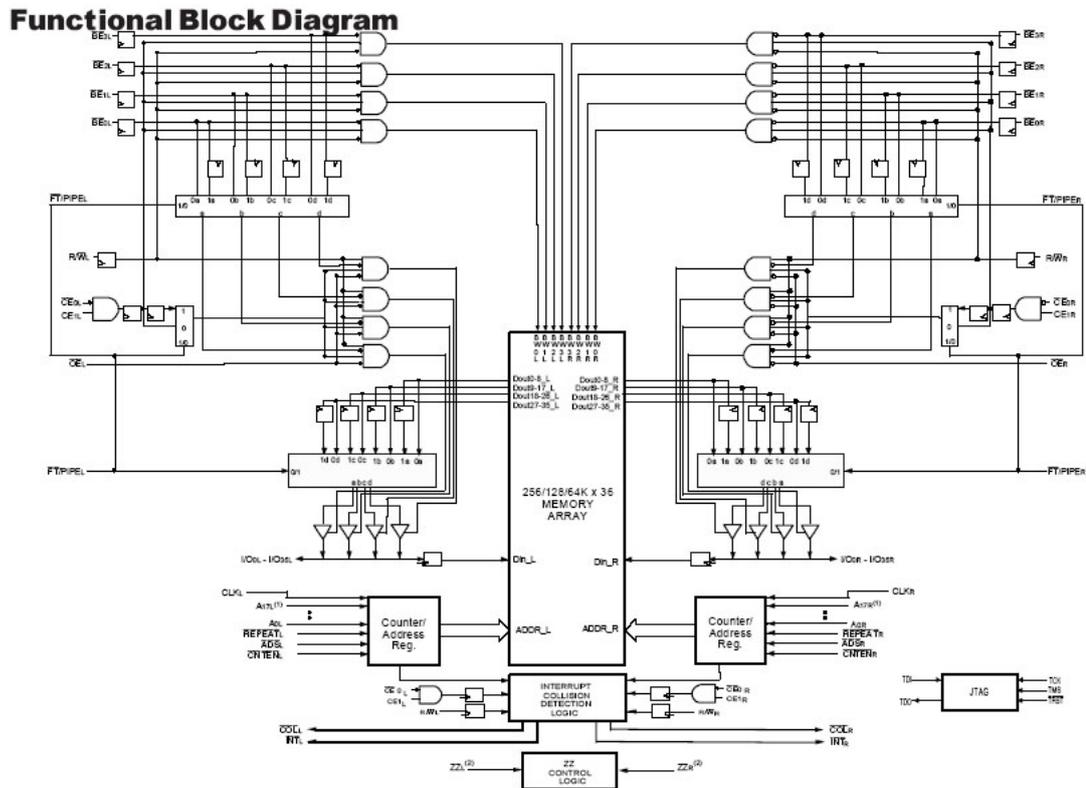


Figura 5.4: Schema a blocchi della memoria Dual-Port IDT70T3519

I dati e gli indirizzi della memoria sono connessi al bus locale dei DSP mentre il segnale \bar{E} di abilitazione è connesso all'FPGA E1C4A. E' quest'ultimo componente quindi che avvia il processo di inizializzazione dei processori. Vi sono diversi possibili schemi di inizializzazione per un sistema multiprocessore costituito da ADSP21160N. Essi possono essere selezionati attraverso i segnali EBOOT, LBOOT e BMS presenti su ogni DSP. Nel nostro caso solo il DSP1 è connesso alla memoria FLASH e l'inizializzazione avviene attraverso le seguenti operazioni in successione:

- l'EP1C4A abilita la memoria FLASH attraverso il segnale \bar{E}
- il DSP1 richiede ed ottiene il controllo del bus locale
- il DSP1 scarica dalla FLASH 256 istruzioni suddivise in parole da 48 bit e le esegue. Queste istruzioni, dette Loader Kernels, sono necessarie per poter caricare altri eventuali dati o istruzioni contenuti nella FLASH.

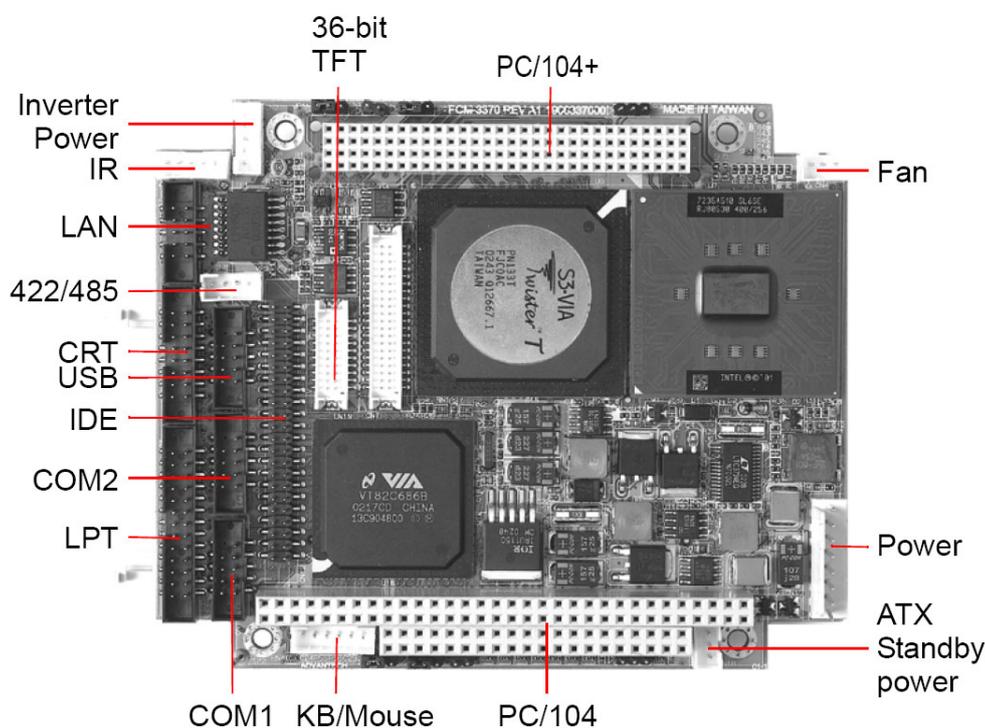


Figura 5.5: Il modulo PC/104Plus Advantech PCM-3370

- il DSP1 scarica le restanti istruzioni contenute nella FLASH e le esegue
- il DSP1 inizializza i DSP2-6 usando lo spazio di memoria multi-processore condivisa.

5.1.4 Modulo PC/104

Nel nostro sistema il ruolo attualmente svolto dalla RIO8062 è ricoperto dal modulo PC/104Plus PCM-3370 della Advantech mostrato in fig. 5.5.

Lo standard PC/104Plus definisce le caratteristiche meccaniche ed elettriche di alcune CPU modulari molto usate in campo industriale. Lo standard prevede delle dimensioni massime di 86x96x15 mm e la presenza di due connettori ISA e di un connettore dedicato al bus PCI. Quest'ultimo supporta solo la versione a 32 bit e 33 MHz del bus al fine di ridurre al minimo le dimensioni. Uno dei vantaggi dei moduli PC/104 è la possibilità di collegare più moduli tra loro, mettendoli uno su l'altro, attraverso i due connettori ISA.

Sul PCM-3370 è installato un processore della famiglia x86, il PENTIUM

III a 933 MHz. Il modulo è inoltre dotato di tutte le caratteristiche di un comune PC commerciale: 512 MB di memoria SDRAM, una scheda video integrata, un'interfaccia ETHERNET 10/100 Mbit, 2 porte USB 1.1, 2 porte seriali.

5.2 Layout del circuito

Terminata la fase di progettazione elettrica è necessario definire le caratteristiche fisiche del circuito stampato. Questa fase della progettazione elettronica è denominata layout. Essa è già stata in parte ultimata. Il software utilizzato è OrCAD Layout prodotto dalla Cadence.

Un moderno PCB (Printed Circuit Board) è costituito da una serie di piani, detti layers, in numero generalmente compreso tra 1 e 20. I componenti vengono saldati su uno o su entrambi gli strati più esterni. In circuiti che, come il nostro, operano a frequenze di clock molto elevate è inoltre necessario utilizzare due o più piani dedicati esclusivamente alla massa (ground plane) e alle alimentazioni (power plane) [27]. Per questo progetto verranno utilizzati 16 strati di cui 9 saranno disponibili per le varie connessioni (routing layers) e 7 saranno dedicati solo a masse ed alimentazioni. I vari layers sono connessi tra loro attraverso dei fori denominati *vias* che vengono realizzati da un laser.

Le fig. 5.6 e 5.7 mostrano la disposizione prevista dei componenti sui due lati della scheda. Il PCB sarà quindi a doppia faccia e con dimensioni di 7 x 14 cm.

La fase finale del layout di un circuito stampato è costituita dalla definizione della geometria ottimale di tutte le varie connessioni, processo denominato *routing*. A tal fine è necessario estrarre dallo schematico elettrico tutte le proprietà dei segnali e riunirle in unico file, detto *netlist*. Partendo dalla *netlist* un software dedicato, chiamato *autorouter*, cerca, attraverso una serie di algoritmi specifici, la configurazione geometrica ottimale di ogni connessione. Il routing della scheda DSP è attualmente ancora in corso.

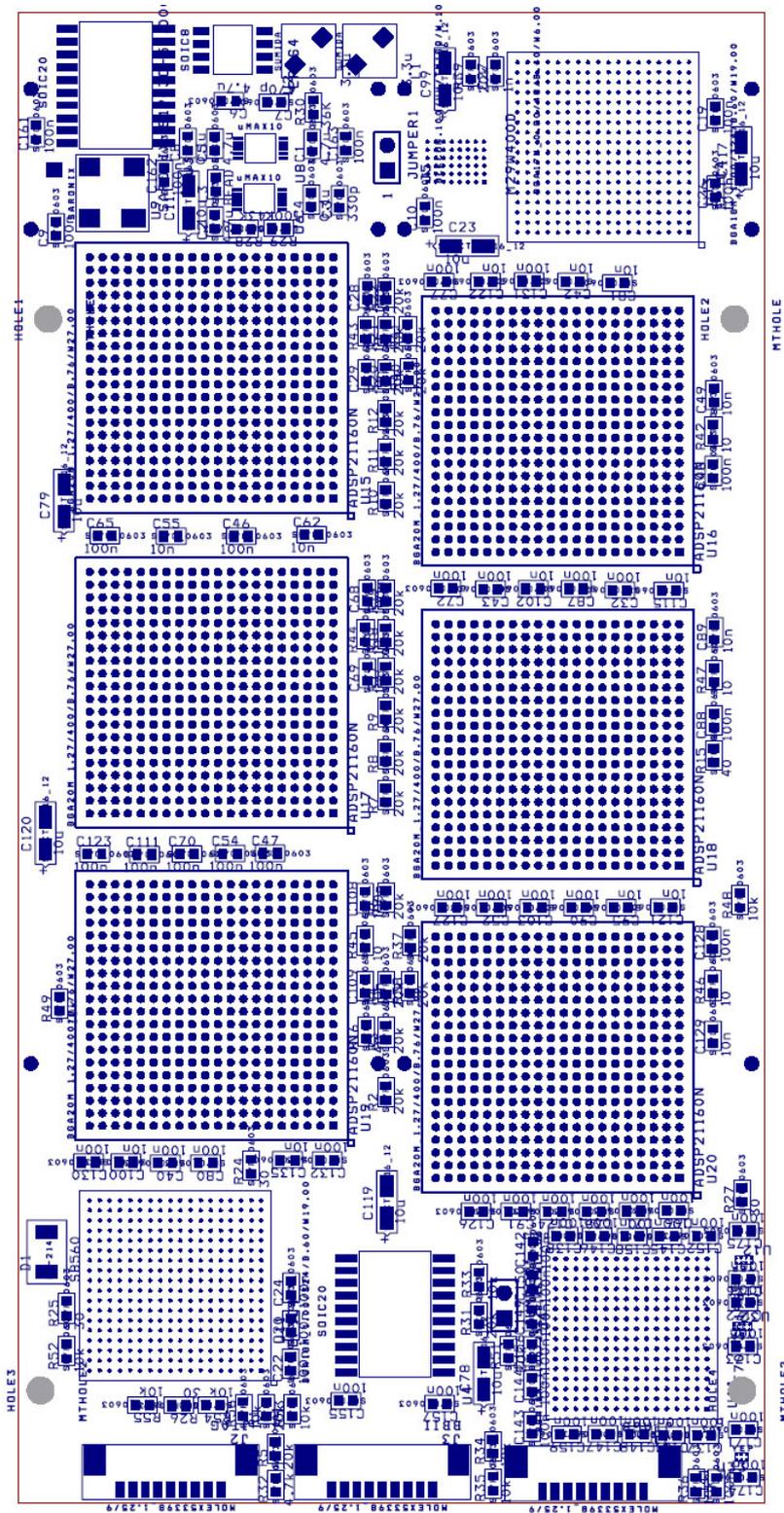


Figura 5.6: *Disposizione dei componenti sulla nuova scheda DSP (lato A)*

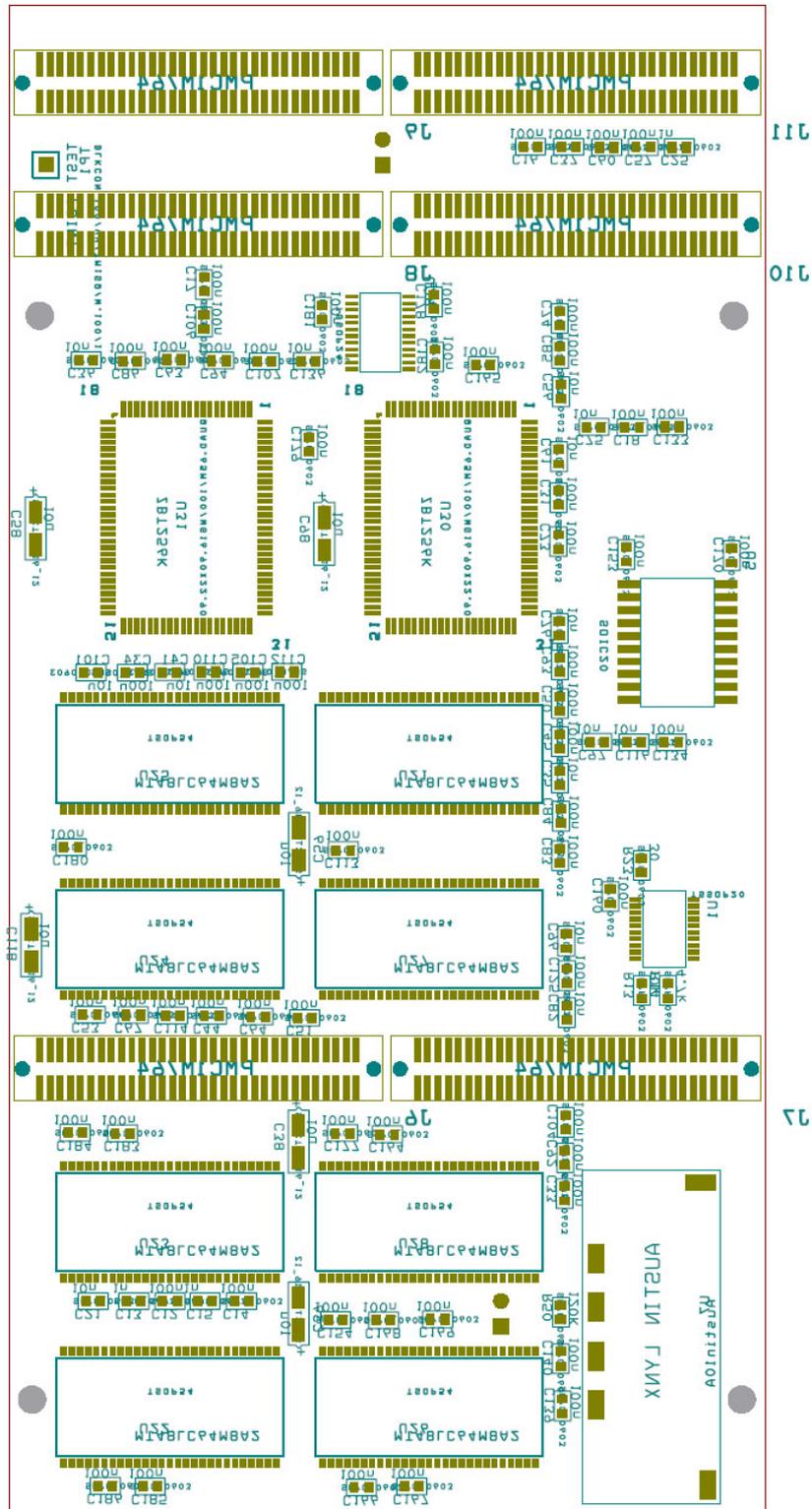


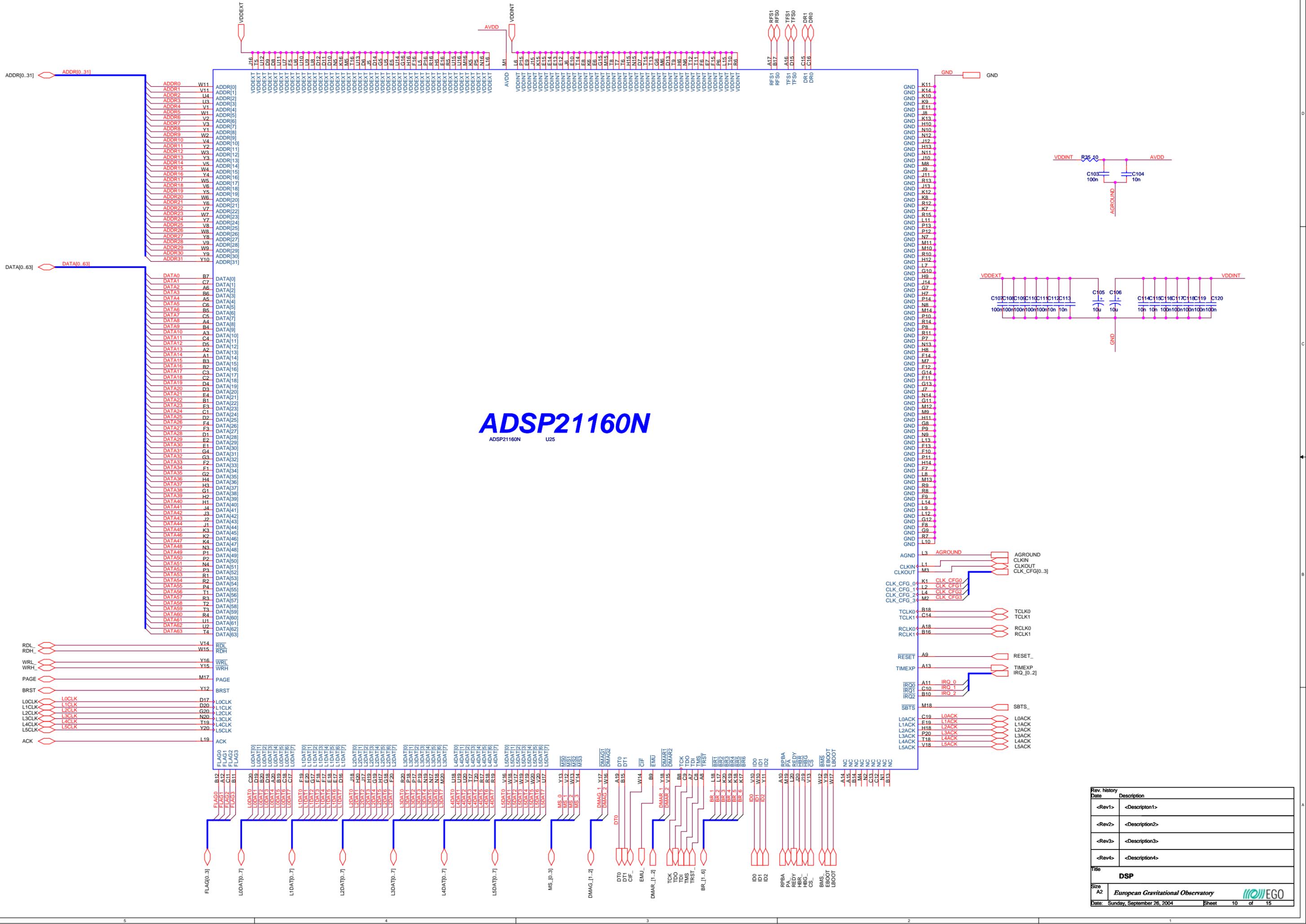
Figura 5.7: Disposizione dei componenti sulla nuova scheda DSP (lato B)

Appendice A

Schemi elettrici

A partire dalla prossima pagina sono riportati gli schemi elettrici del progetto. L'ordine degli schematici è il seguente:

1. Schema generale del circuito.
2. L'ADSP21160N
3. La FPGA Cyclone EP1C4A
4. La FPGA Cyclone EP1C4B
5. Il banco di memoria SDRAM
6. Il banco di memoria ZBT
7. La memoria Dual-Port
8. Connettori PMC slot A
9. Connettori PMC slot B



ADSP21160N

ADSP21160N U25

Rev. history	
Date	Description
<Rev1>	<Description1>
<Rev2>	<Description2>
<Rev3>	<Description3>
<Rev4>	<Description4>

Title: **DSP**

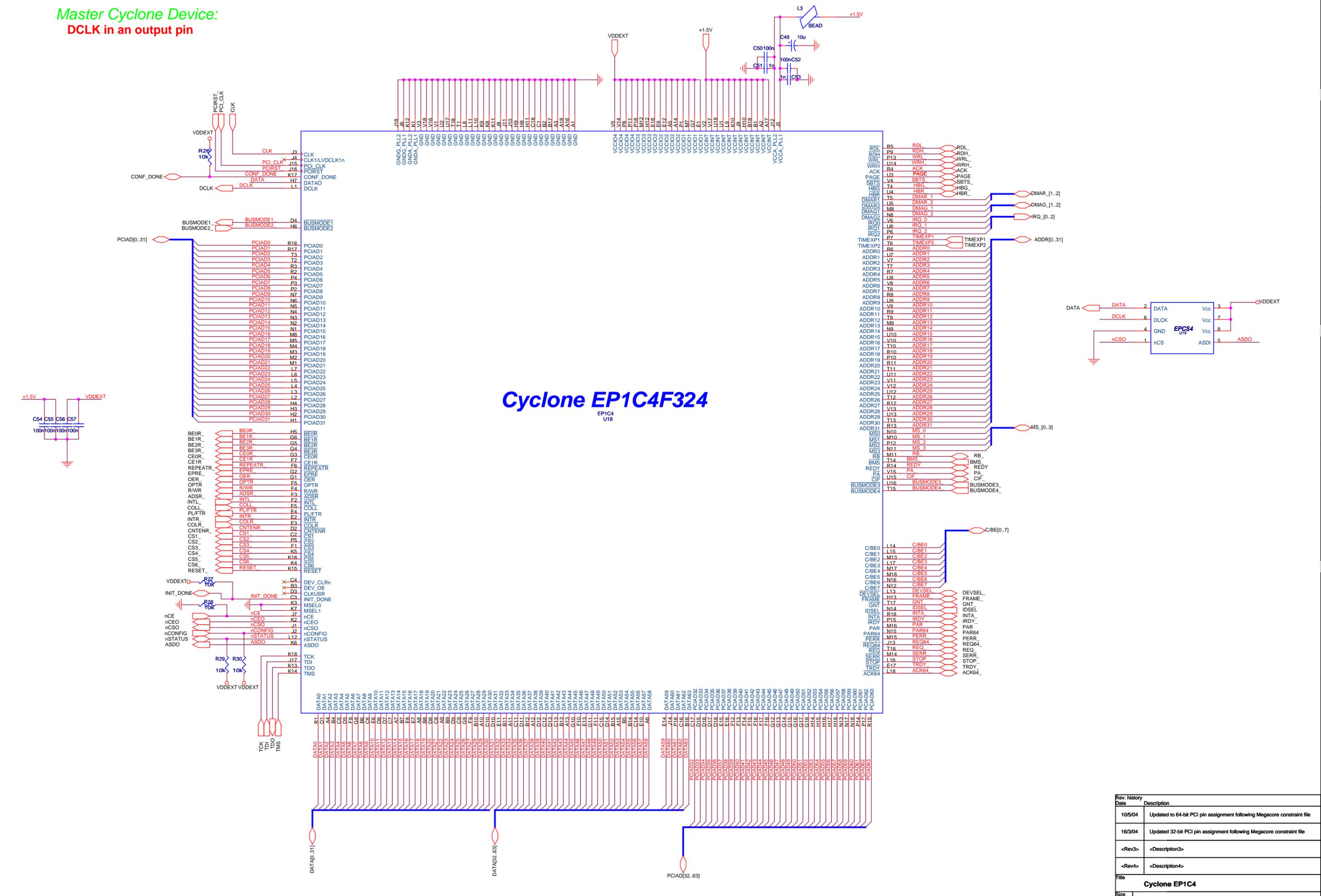
Size: **A2**

Date: Sunday, September 26, 2004

Sheet 10 of 15



Master Cyclone Device:
DCLK in an output pin



Cyclone EP1C4F324
EP1C4 U18

Rev. history	Date	Description
	10/5/04	Updated to 64-bit PCI pin assignment following Megacore constraint file
	16/3/04	Updated 32-bit PCI pin assignment following Megacore constraint file
<Rev3>	<Description3>	
<Rev4>	<Description4>	

Title
Cyclone EP1C4

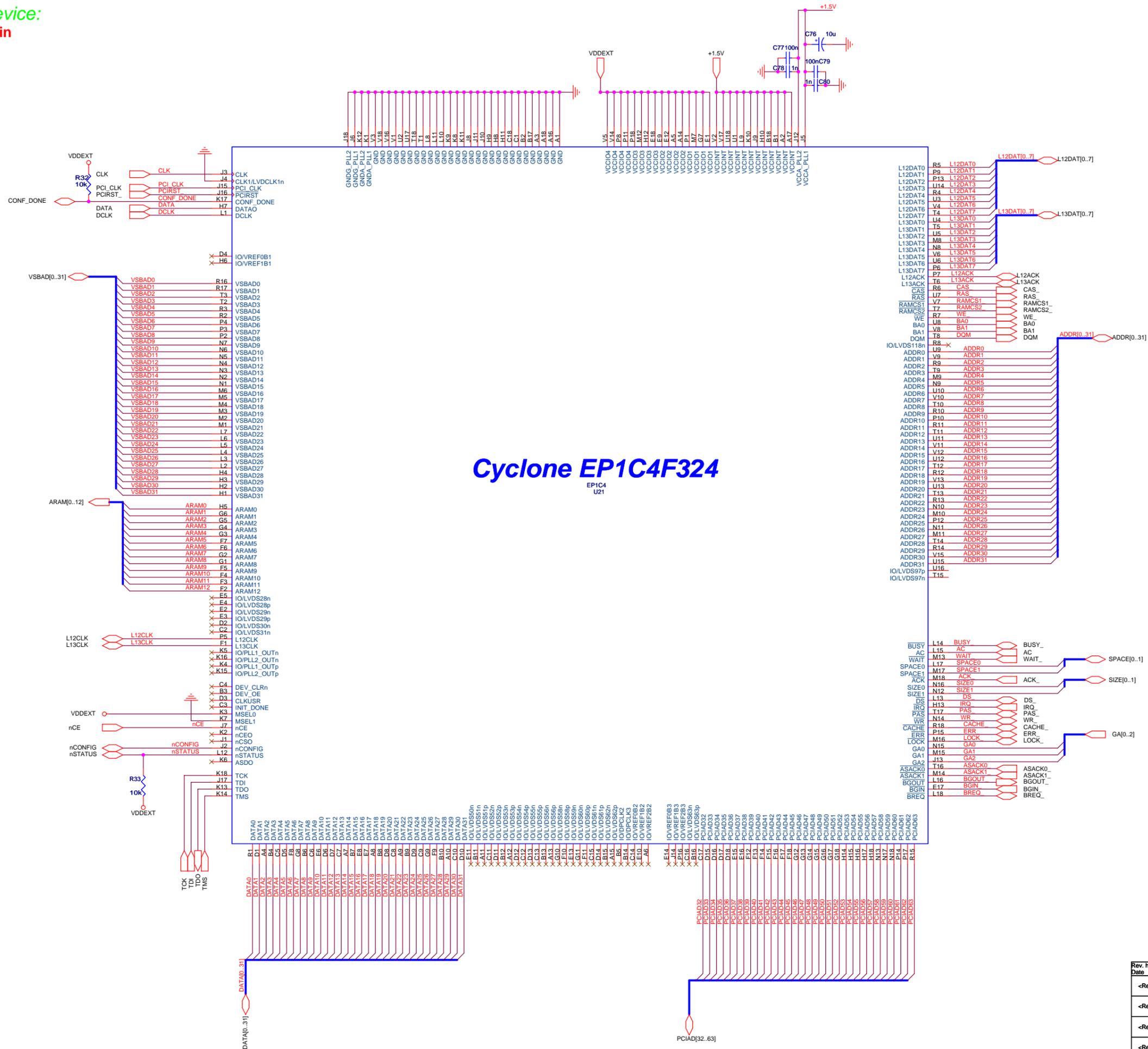
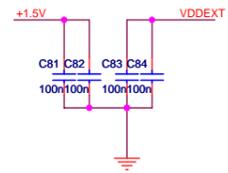
Size
A2

Date: Tuesday, October 12, 2004

Sheet 4 of 15

European Gravitational Observatory EGO

Slave Cyclone Device:
DCLK in an input pin



Cyclone EP1C4F324

EP1C4
U21

Rev. history	
Date	Description
<Rev1>	<Description1>
<Rev2>	<Description2>
<Rev3>	<Description3>
<Rev4>	<Description4>

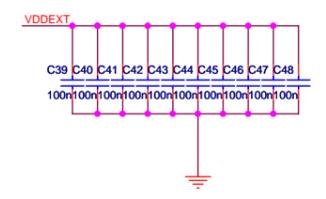
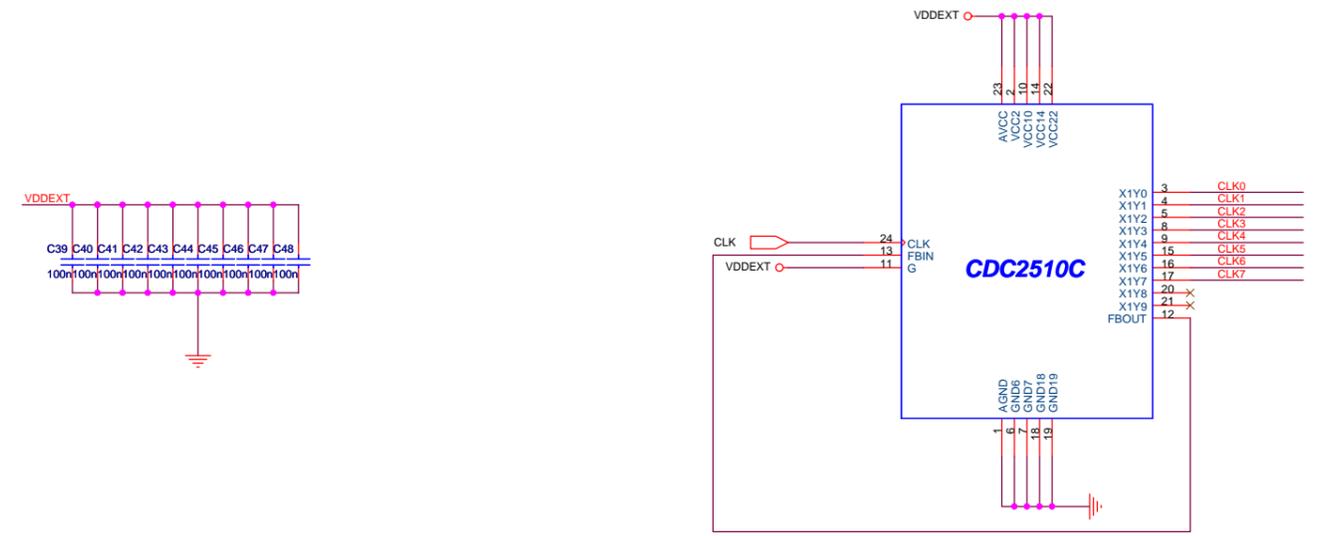
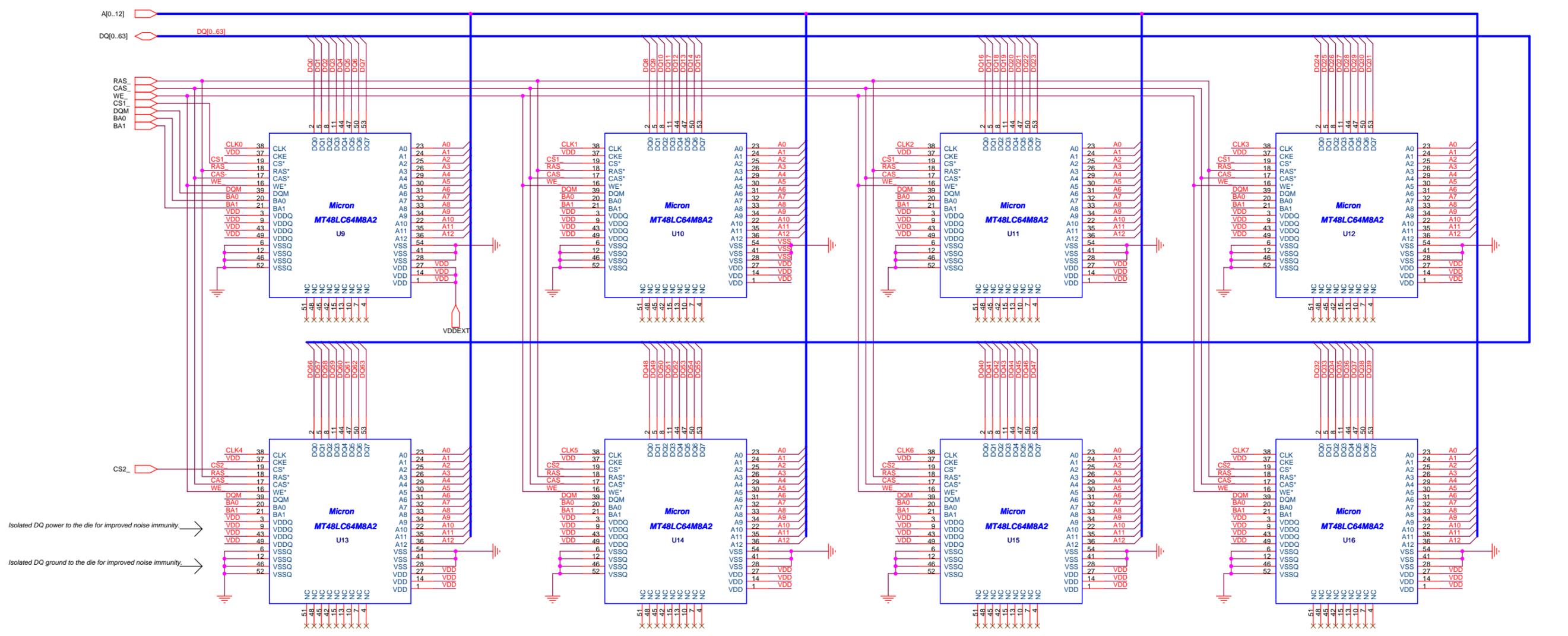
Title
DDR controller + VSB bridge

Size
A2

Date: Tuesday, October 12, 2004

Sheet 6 of 15





Rev. history	
Date	Description
<Rev1>	<Description1>
<Rev2>	<Description2>
<Rev3>	<Description3>
<Rev4>	<Description4>

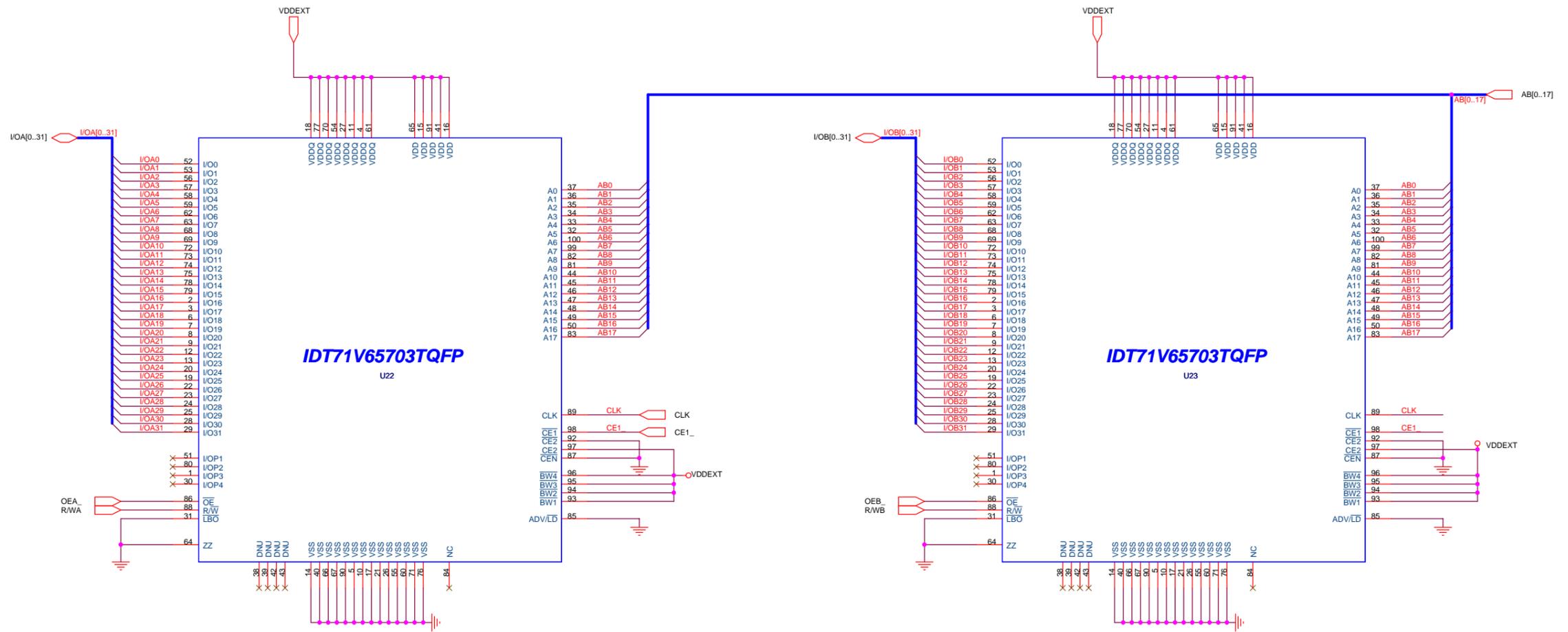
Title
512MB SDRAM memory banks

Size
A2

Date: Sunday, September 26, 2004

Sheet 3 of 15

European Gravitational Observatory



Rev. history	
Date	Description
<Rev1>	<Description1>
<Rev2>	<Description2>
<Rev3>	<Description3>
<Rev4>	<Description4>

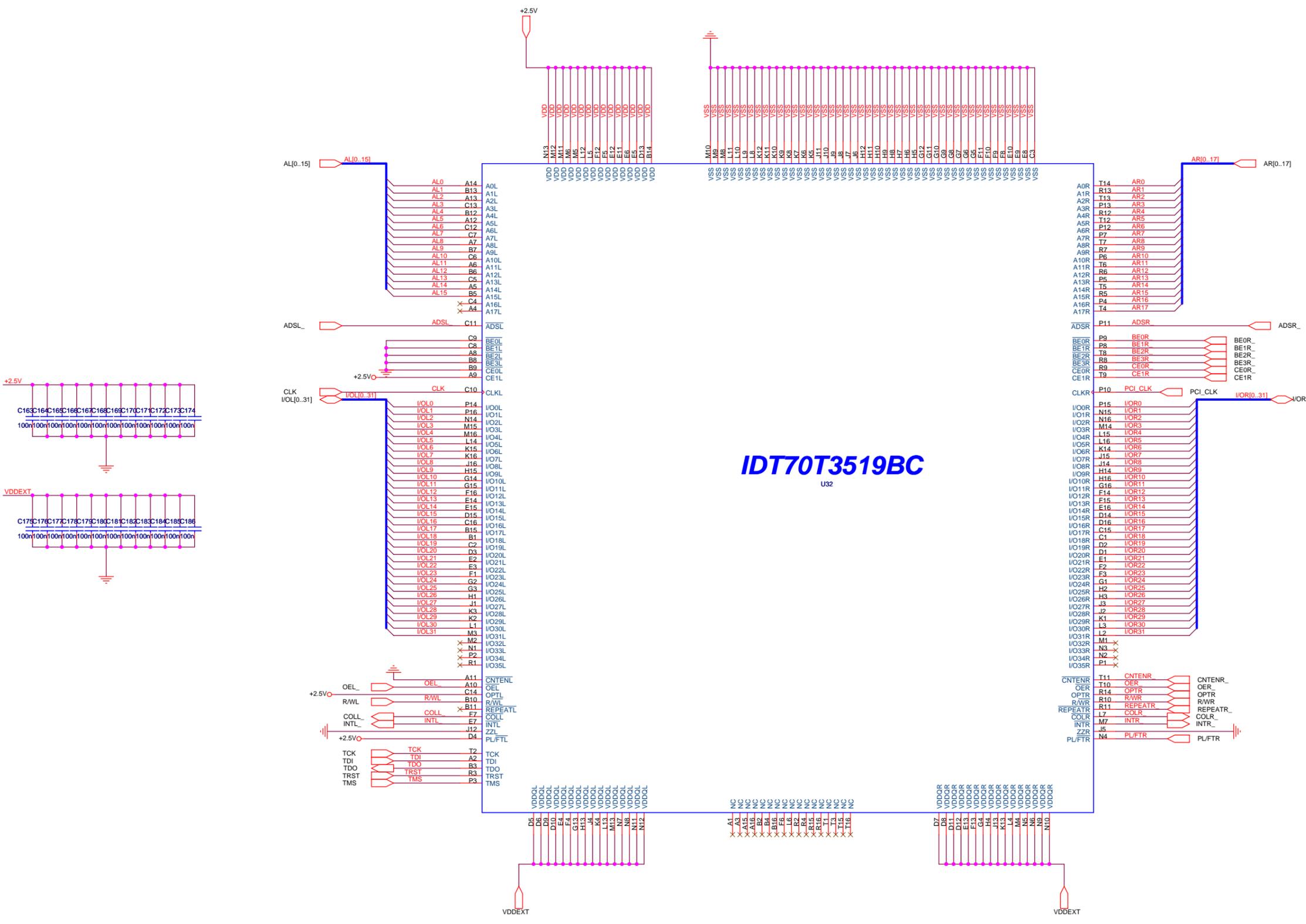
Title
ZBT Memory banks

Size
A2

Date: Wednesday, October 13, 2004

Sheet 7 of 15





IDT70T3519BC
U32

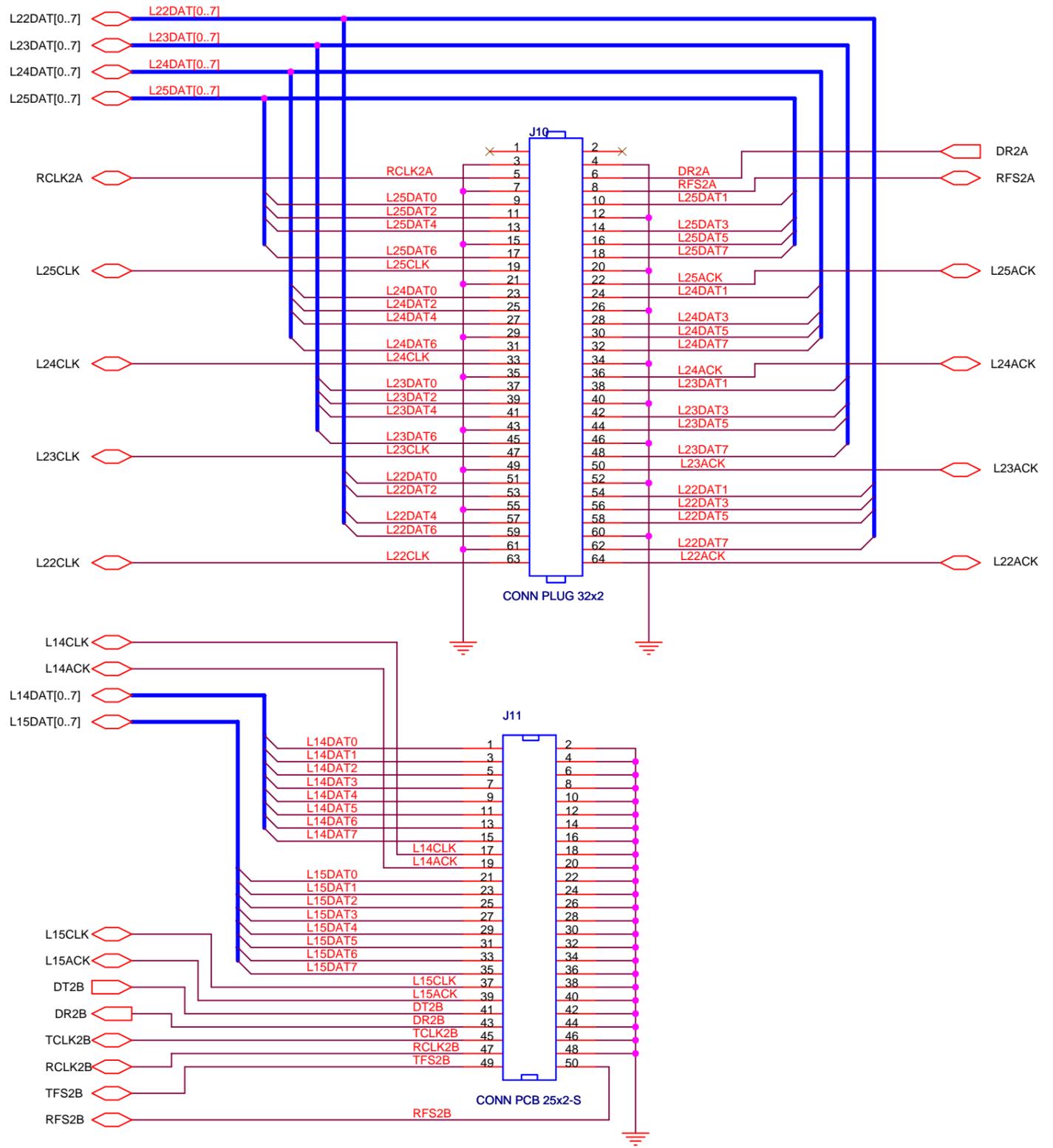
Rev. history	
Date	Description
<Rev1>	<Description1>
<Rev2>	<Description2>
<Rev3>	<Description3>
<Rev4>	<Description4>

Title
IDT Dual Port RAM

Size
A2

Date: Sunday, September 26, 2004 Sheet 15 of 15

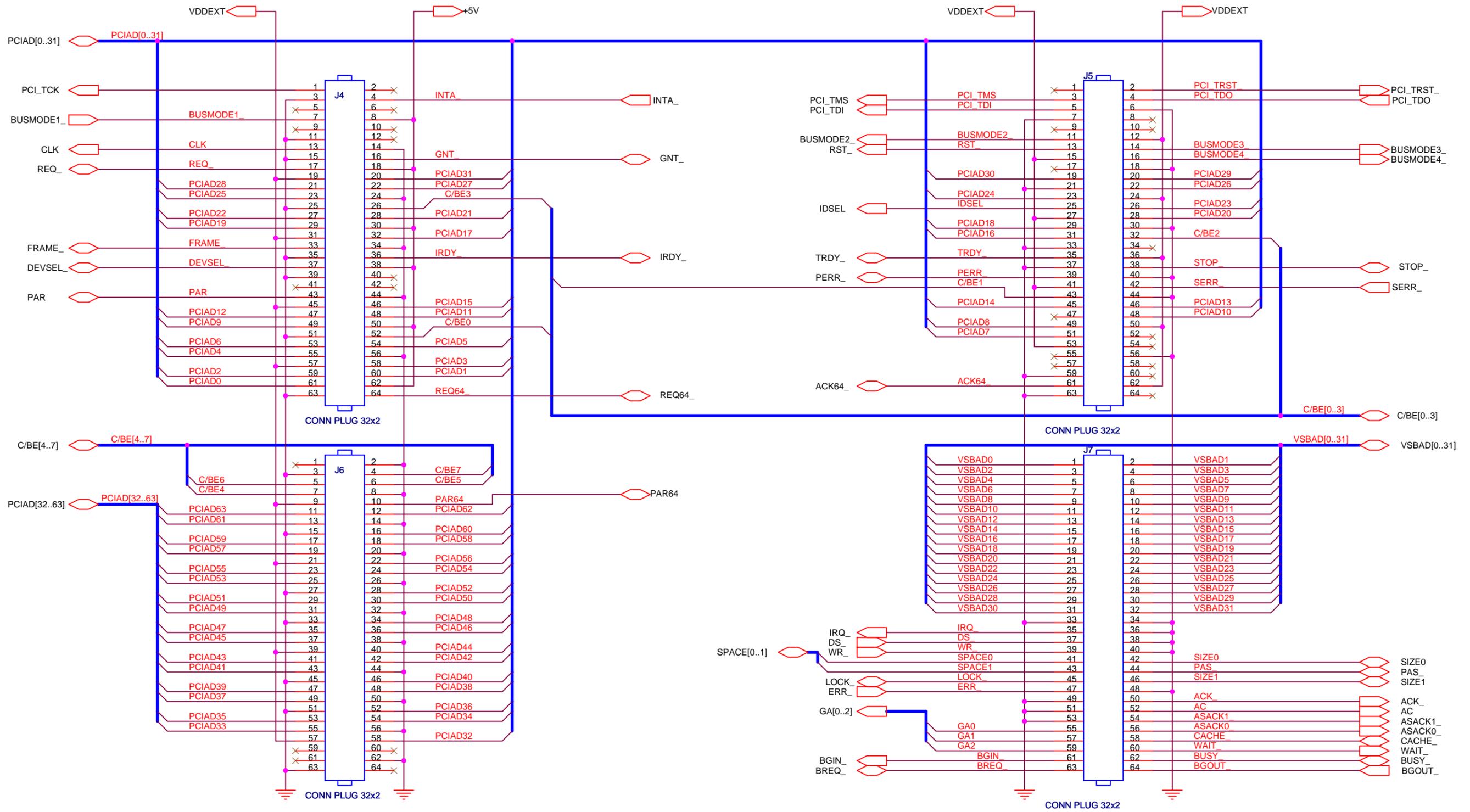
European Gravitational Observatory EGO



Rev. history	Description
Date	
<Rev1>	<Description1>
<Rev2>	<Description2>
<Rev3>	<Description3>
<Rev4>	<Description4>

Title
2 x PMC CONNECTORS

Size A3
European Gravitational Observatory 



Rev. history	
Date	Description
<Rev1>	<Description1>
<Rev2>	<Description2>
<Rev3>	<Description3>
<Rev4>	<Description4>

Bibliografia

- [1] S.Weinberg, *Gravitation and cosmology*, Wiley, New York (1993)
- [2] P.R. Saulson, *Fundamentals of interferometric gravitational wave detectors*, World Scientific(1994)
- [3] T.Zwenger, E.Muller, *Astron. Astrophys.*, 320 209 (1997)
- [4] R. H. Dicke, *The Theoretical Significance of Experimental Relativity*, Gordon and Breach, New York (1964)
- [5] J. Weber, *Phys. Rev. Lett.* 22 (1969) 1320
- [6] F. Cavalier, *Le contrôle global de Virgo*, LAL 01-69
- [7] C. Casciano, *Seismic isolation for the test masses of the Virgo gw antenna*, tesi di dottorato in fisica (2002)
- [8] G. Losurdo, *Ultra low frequency inverted pendulum for the Virgo test mass suspension*, tesi di perf. in fisica (1998).
- [9] S.Braccini et al., *Low noise wideband accelerometer using an inductive displacement sensor*, *Rev. Sci. Instrum.* 66, 2672 (1995)
- [10] G. Marro, *Controlli automatici*, Zanichelli (1997)
- [11] J.M.Maciejowski, *Multivariable Feedback Design*, Addison-Wesley (1989)
- [12] A. V. Oppenheim, R. W. Shafer, *Digital Signal Processing*, Prentice-Hall (1975)
- [13] K. Ogata, *Discrete-time control systems*, Prentice-Hall (1995)
- [14] D. Passuello, *Filtri digitali*, non pubblicato
- [15] *IEEE standard for binary Floating-Point arithmetic*, IEEE Std 754-1985
- [16] Analog Devices, AD976/AD976 Datasheet, (1999)

- [17] T. Kugelstadt, *The operation of the SAR-ADC based on charge redistribution*, Texas Instruments Analog Application Journal, 2/2000
- [18] R. Taddei, *Progetto e realizzazione di un sistema digitale per il controllo hard real time di sistemi multivariabile per l'esperimento Virgo*, tesi di laurea in ingegneria elettronica, (1996)
- [19] Motorola, *DSP96002 User's Manual*, (1996)
- [20] CES, *RIO2 8062 PowerPC based RISC I/O Board Manual*, (1998)
- [21] D. Boget et al., *Virgo Note VIR-SPE-LAP-5200-105* (1998)
- [22] G. Buttazzo, *Sistemi in tempo reale*, Pitagora Editrice, Bologna (1995)
- [23] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, San Diego (1999)
- [24] Analog Devices, *ADSP-21160 SHARC DSP Hardware Reference*, (2002)
- [25] *PCI Local Bus Specification rev 2.2*, PCISIG (1998)
- [26] *Cyclone Device Handbook*, Altera (2003)
- [27] H. Johnson, M. Graham, *High-Speed Digital Design*, Prentice-Hall (1993)

Ringraziamenti

Ringrazio Francesco Fidecaro che mi ha introdotto all'esperimento e alle varie problematiche della gravitazione sperimentale, Diego Passuello che ha supervisionato il mio lavoro ed ha risposto ad ogni mio dubbio e problema; la sua profonda competenza in ogni campo della ricerca scientifica rende ogni discussione con lui un'occasione di arricchimento culturale.

Ringrazio Alberto Gennai, punto di riferimento costante in ogni fase del progetto, Roberto Taddei, che mi ha introdotto ai vari aspetti della progettazione elettronica; il suo valore umano e la sua grande simpatia hanno reso piacevole ogni momento trascorso a Virgo.

Un ringraziamento va anche a Giulio Ballardin, Federico Paoletti, Christoff Moins e al resto del gruppo elettronico di Virgo che hanno contribuito a rendere il mio ambiente di lavoro estremamente rilassato e divertente.

Desidero inoltre ringraziare tutte le persone che, fuori e dentro il dipartimento, mi sono state vicine durante il corso di laurea: Simone, Daniele, Gian Maria, Andrea, Paolo, Emiliano, Valerio, Davide e molti altri.

Last but not least, ringrazio i miei genitori senza cui certamente questo lavoro non sarebbe stato possibile.