



Corso Luigi Einaudi, 55 - Torino

Appunti universitari

Tesi di laurea

Cartoleria e cancelleria

Stampa file e fotocopie

Print on demand

Rilegature

NUMERO: 1313

ANNO: 2014

A P P U N T I

STUDENTE: Beccaria

MATERIA: Sistemi Digitali Integrati, Prof.Ruo Roch

Il presente lavoro nasce dall'impegno dell'autore ed è distribuito in accordo con il Centro Appunti.

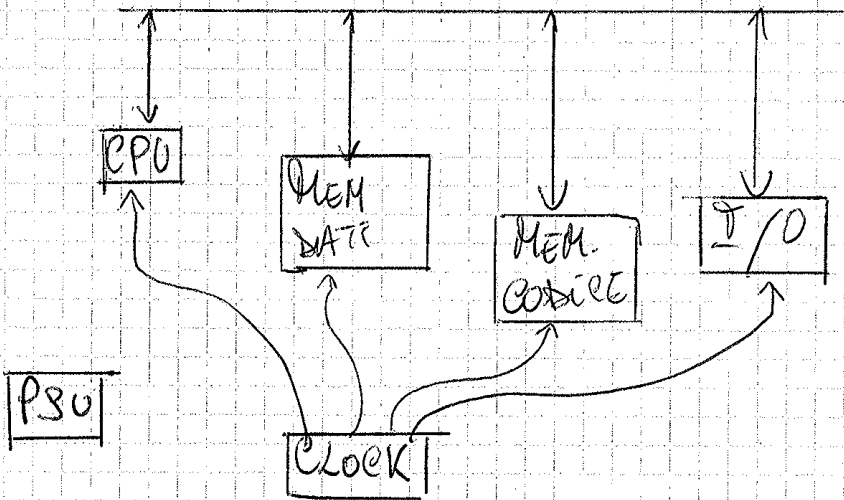
Tutti i diritti sono riservati. È vietata qualsiasi riproduzione, copia totale o parziale, dei contenuti inseriti nel presente volume, ivi inclusa la memorizzazione, rielaborazione, diffusione o distribuzione dei contenuti stessi mediante qualunque supporto magnetico o cartaceo, piattaforma tecnologica o rete telematica, senza previa autorizzazione scritta dell'autore.

ATTENZIONE: QUESTI APPUNTI SONO FATTI DA STUDENTIE NON SONO STATI VISIONATI DAL DOCENTE.
IL NOME DEL PROFESSORE, SERVE SOLO PER IDENTIFICARE IL CORSO.

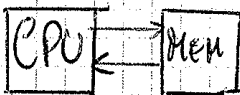
30/09/2013

①

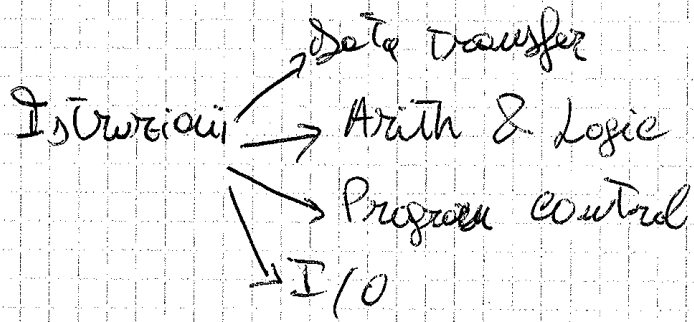
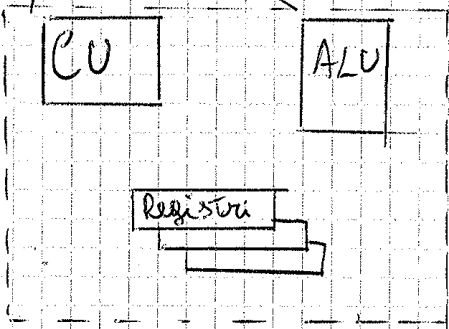
System embedded



Architecture Von-Neumann



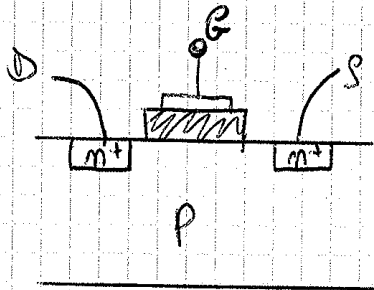
La CPU interagisce con una memoria



Si utilizza per memorizzare dei dati che non cambieranno mai o del codice per esempio. (3)

→ Memorie a sola lettura (Mostly Read)

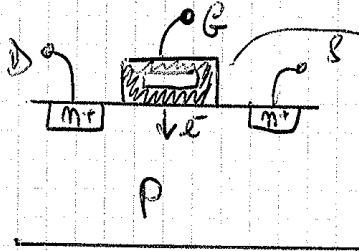
Si usa il transistor. Ma MOS a gate flottante



transistor ad arricchimento

Se applico tensione sul gate ^{il MOS} diventa conduttivo se la V non supera la V_{th} non si ha passaggio di corrente.

Se si modifica la V_{th} così:

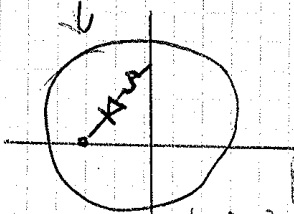


riesco ad accumulare cariche sul gate facendo variare la V_{th}

Si applica un campo E tra D e S , gli elettroni sono accelerati, e a causa dei moti esso viene deflessi può essere anche deflesso a 90° , e si parla di iniezione di elettroni caldi, modificando così la tensione di soglia

• PROVA

modo con un sonda a fusibile



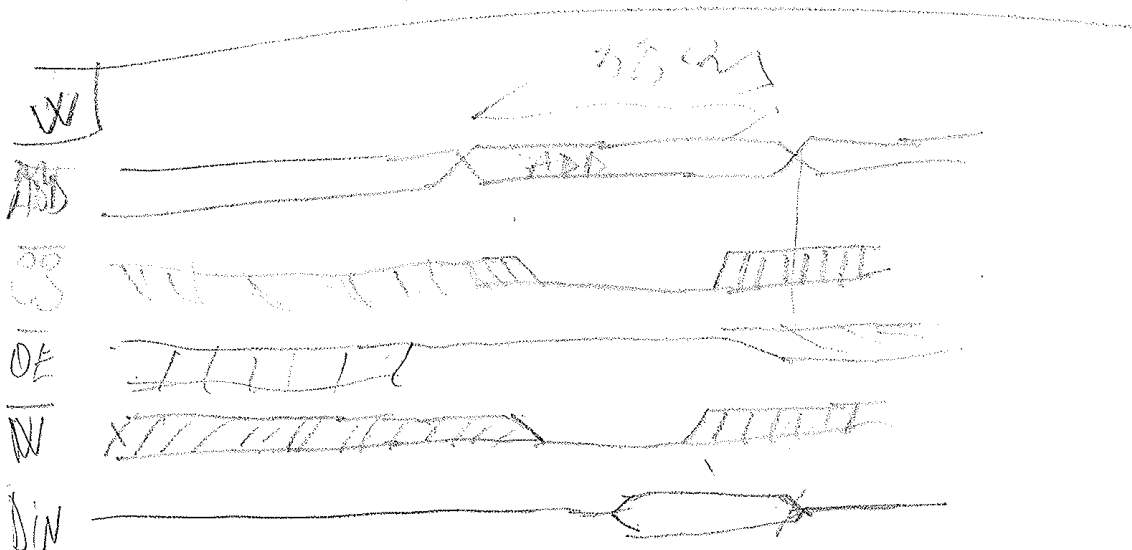
↑ scrittura geometricamente il effetto tunnel dice che: poiché la penetrazione di un portatore non è rappresentata da un coordinato $N-y$ ma da una densità di probabilità (fuori Schrodinger), quando dico che un portatore è qui non vuol dire il port. è qui vuol dire che ha la max probabilità di essere qui un po' a destra o a sc. Se io metto una barriera di potenziale sottile e piuttosto a portare un portatore vicino in realtà dal p.to di vista geometrico è un po' oltre la porta della barriera, rispetto con dei campi elettrici a farli entrare con il tunnel

Il vantaggio è il minor costo, lo svantaggio è la tecnologia più complicata e di conseguenza il costo per testarli sono maggiori. Problemi di affidabilità perché c'è

Mem. FLASH

(5)

Sono mem. F^2 PROM, aggiungendo delle limitat. sull'utilizzo sono più dense, e perità di come ci stanno più bit, ~~con~~ limitazioni sulla programmazione. ~~Lo~~ ~~Resti~~ ~~sono~~ Viene introdotto un cross-trade cioè quando faccio una scrittura posso sempre scrivere una parola (come l' E^2 PROM) ma se faccio una cancellazione non ~~posso~~ cancellare la singole parole ma solo un insieme di parole che si chiama blocco. (max per 32KB) (scrittore come E^2 PROM, cancella no!) Se si vuole modif. una word si deve salvare in RAM il blocco, cancellarlo e poi riscriverlo! Si trovano ovunque del p.to di vista del tempo. Però la stabilità è migliore. Il vantaggio è il WRITE ENDURANCE è aumentato fino ad un milione. Nascono così gli SSD (scrittore limitate)

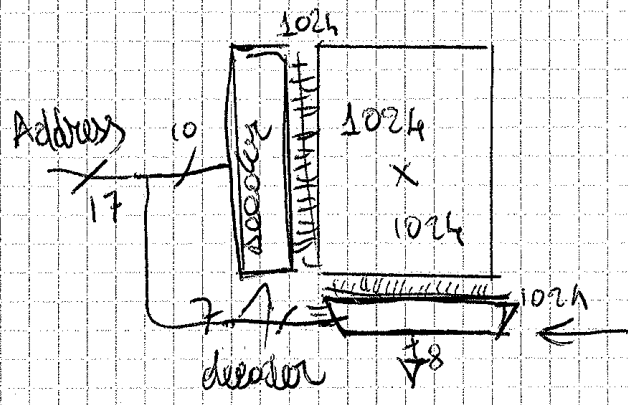


Le capacità forza il latch e cambiare stato. Per evitare questo si usa la tecnica della pre-carica (mos in alto con $V_{cc}/2$) quando accendo la W-line. . . .

(7)

Si utilizza il comparatore

Si vuole costruire una memoria 128Kx8



si inserisce il mux per avere l'8

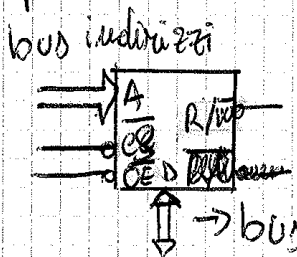
Quando si attivano i pass transistor, le capacità forza il latch e cambiare stato. Tecnica di pre-carica: se c'è uno 0 o un 1 sulle bit line prima di aprire il pass transistor quando li apre vince la bit-line, allora metto un segnale debole, verde e pre-carico la linea mediante un mos, ad esempio, e lo attaco a $V_{cc}/2$, prima di fare la lettura attivo quei 2 mos di sopra, si mette sul gate un segnale di pre-charge, e portiamo quelle due linee a V_{cc} ; le capacità delle bit-line mantengono $V_{cc}/2$; quando accendo la word-line

Quando attivo i pass transistor, sappiamo che per linee (bit line) sia allo zero logico e dentro il latch c'è un 1 logico, quando chiudo il pass transistor mi aspettavo di vedere l'1 che esce fuori, in realtà le capacità che sul transistor funziona come un bottone di ferro forza il latch e cambiare stato, abilitando il pass transistor. Perché esca un 1 entra un zero (perché le capacità è così forte più di mos riuscendo a invertire lo stato). Se uso la pre-carica, se sulla bit line c'è 0 o 1, prima di aprire i pass transistor quando li apro vince la bit line, metto un segnale debole pre-caricando la linea attacco a $V_{cc}/2$. Prima di fare la lettura attivo i due mos sopra, e portiamo le linee a V_{cc} . Quando accendo la word line se c'è un 1 scritto non scrive nel latch. Non faccio l'errore di un po' tanto tempo per le capacità di tempo

07/10/2013

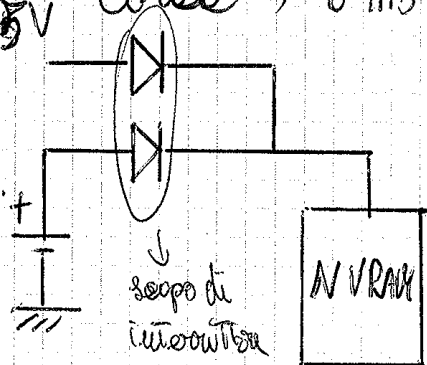
9

La maggior parte di SRAM ha 3 segnali di controllo.



\overline{CS} = chip select - voglio usare la ram
 \overline{OE} = output enable - quando attivo
 R/\overline{WR} = read/write autorizzo la RAM a essere il bus

L'ordine di grandezza di ~~una~~ accesso ad una memoria SRAM è delle 10^{-10} s (con alcune tecnologie si può arrivare a circa 5-6 ns a scapito dell'consumo)

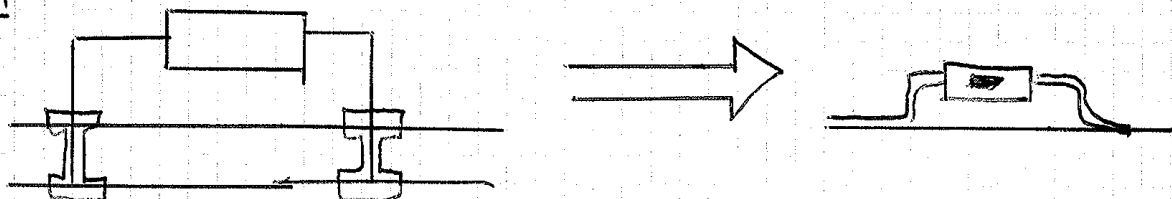


Non-Volatile-RAM

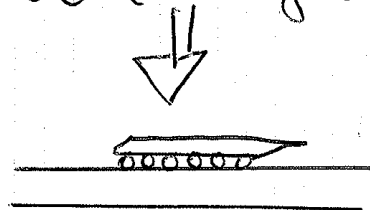
In queste memorie si usa la batteria da appoggio perché se non c'è più la Vcc la mem. mantiene il dato. I componenti si sono evoluti e vedersi montati sulle breadboard con una tecnologia PTH (Plated Through Hole), ogni piedino dista 2,5 mm pertanto tale tecnologia viene soppiantata dalle SMT (Surface Mount) così si ha uno spazio di 0,5 mm, e per un 10^2 di piedini si riduce l'area di un fattore 25

PTH

SMT



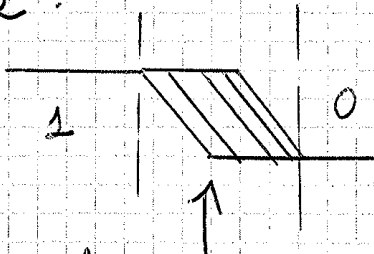
Si è poi passati ad una nuova tecnologia in cui non si utilizza solo la periferia del componente, ma si sfrutta la superficie. (Ball grid Array)



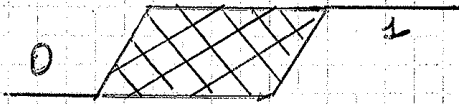
BGA

Poi si ha:

(11)

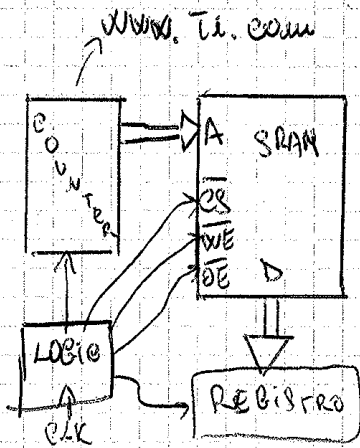


Significa che fa UNA e UNA sola transizione nell'intervallo (non si sa quando esattamente)



Per questo caso si hanno N esecuzioni nell'intervallo (per esempio in un percorso combinatorio)

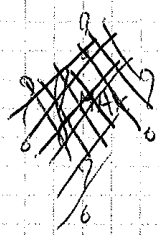
Esercizio



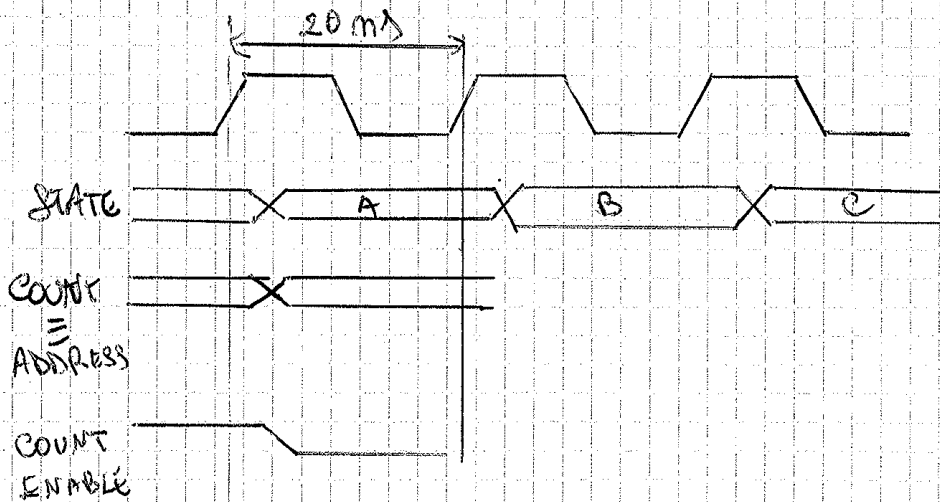
www.ti.com

Tempo di ciclo = 15 ns

$f_{clk} = 5 \text{ MHz}$



$S_{MAX} = \text{quante letture faccio al secondo ???}$



II) Ho \bar{E} aumentata la latenza. 😞 (13) con i registri si deve affrettare il tempo che passa per un dato
 Un altro parametro da considerare è il Through-put (le bande). In questo caso posso cambiare indirizzo ad ogni colpo di clock
 quanti M/s fluiscono nel sistema

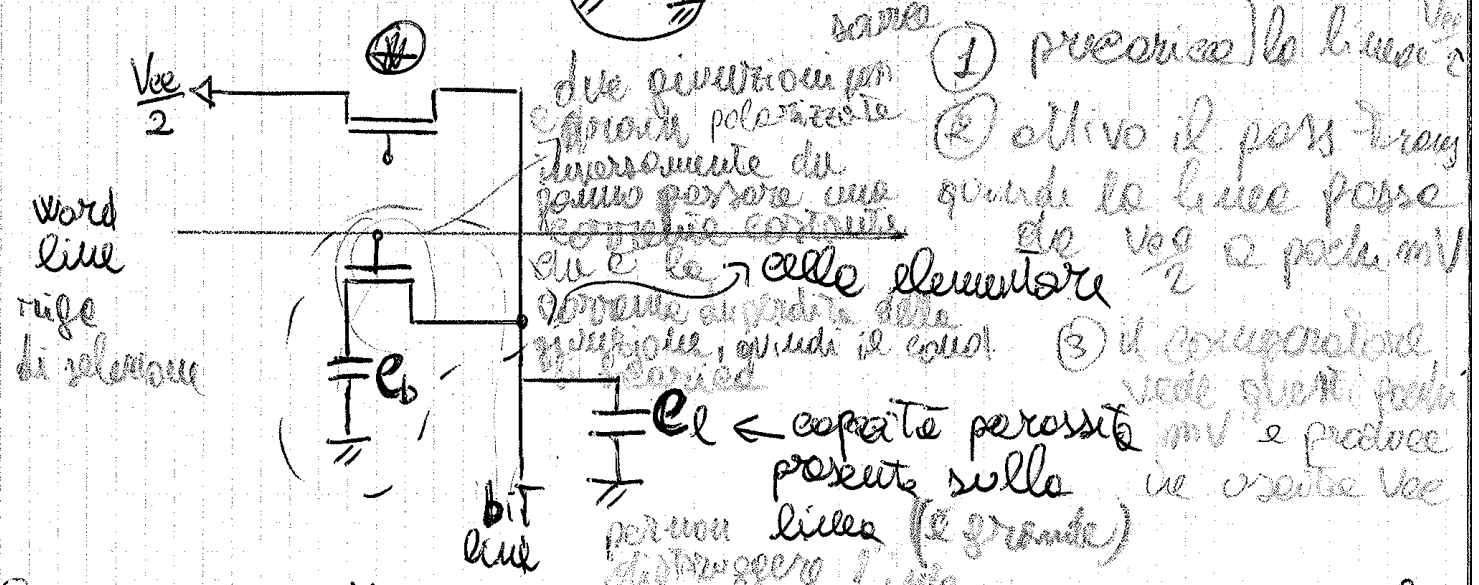
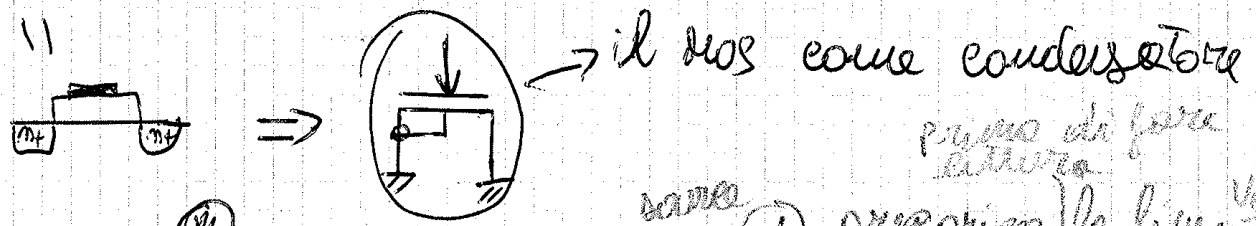
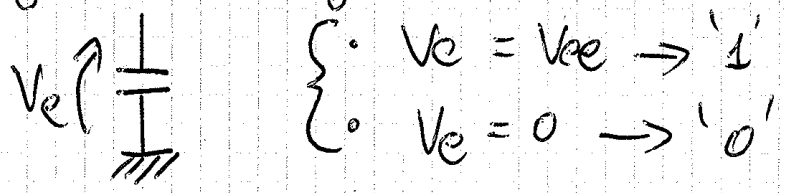
III) Bande ↑↑↑ 😊 (è aumentata la frequenza)

Non sempre è conveniente ricorrere alle pipeline. Ricorrere nei casi in cui conviene tenere le pipeline piene (invio dei dati uno dietro l'altro).

- Vedendo un datasheet di una memoria "pipelinata" si nota che assorbe molta corrente (ex $\approx 0,7A$) e lavora ad altissime frequenze ($\approx 300MHz$)

DRAM

Qual'è il più semplice componente in grado di immagazzinare informazione!

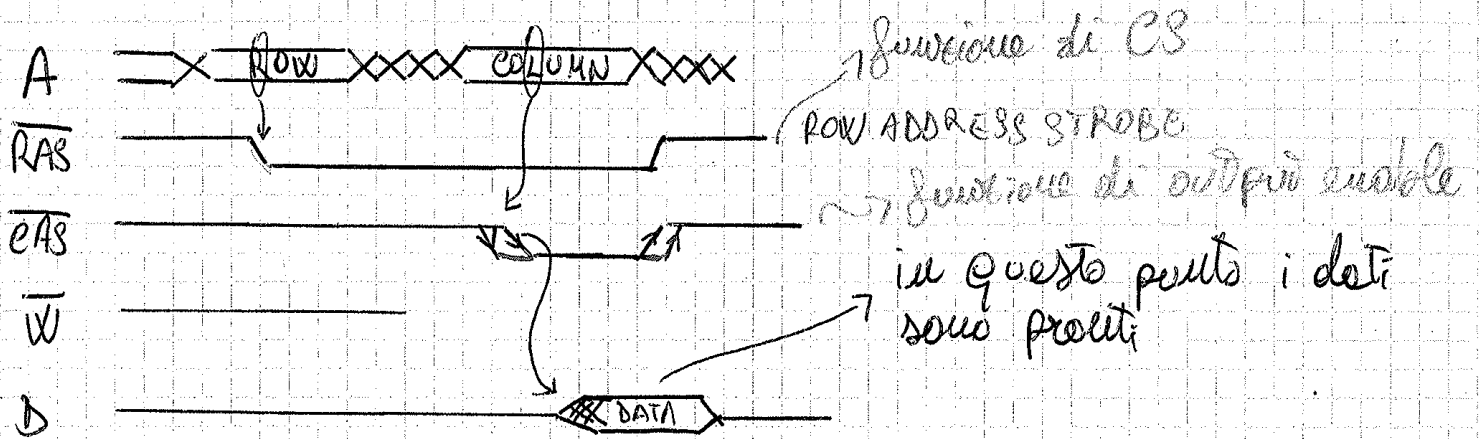


Se attivo la $V_{cc}/2$ wordline la tensione del condensatore si inverte sulla

Segnali di Controllo: A , \overline{RAS} , \overline{CAS}

(15)

• Tecnica di riduzione per di scelte indirizzo: MULTIPLEXING: riduco il n° di bit (di indirizzo) poiché devo selezionare prima la riga e poi la colonna.



Le RAM dinamiche non hanno il chip select (è il \overline{RAS} a fare da chip select). \overline{RAS} e \overline{CAS} sono segnali di strobe. Il \overline{CAS} ha anche funzione di OUTPUT ENABLE. Per quanto riguarda il consumo è praticamente simile alla SRAM.

* Solitamente esistono delle misure di alcuni tempi che sono presi al 90% o al 10% della salita (o discesa) *

Quanti cicli di banda mi porta via il refresh?

Ex. 1024 Refresh / 16 ms

$T_{acc} = 55 \text{ ns}$

\downarrow
~ 1000

\rightarrow tempo accesso RAM

1 Refresh / 16 μs

NB: un ciclo di refresh dura esattamente una lettura.

Supponiamo che $T_{READ} = T_{REFRESH} = 100 \text{ ns}$

In 16 μs \rightarrow 160 letture (160 slot) $\frac{16 \cdot 10^{-6}}{100 \cdot 10^{-9}} = 160$

Ogni 160 "buchi" ne perdo uno per fare il refresh:

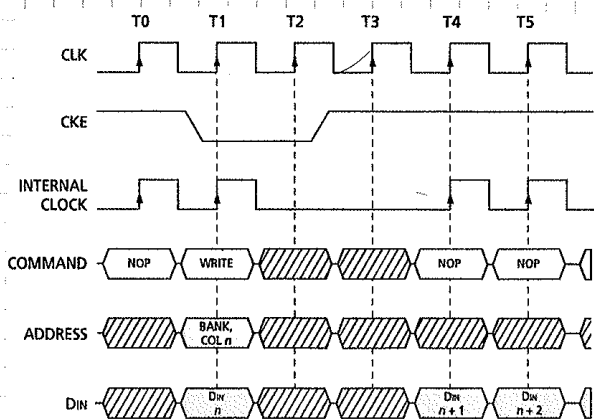
$$\frac{1}{160} \cdot 100 =$$

$< 1\%$

\rightarrow del tempo impiegato

Quanto è la lettura di una lettura? 100 ns se un va bene
0,200 ms (se voglio leggere mentre sta facendo il refresh) \checkmark
del 100% \rightarrow INDETERMINAZIONE DELLA PIÙ SORSA

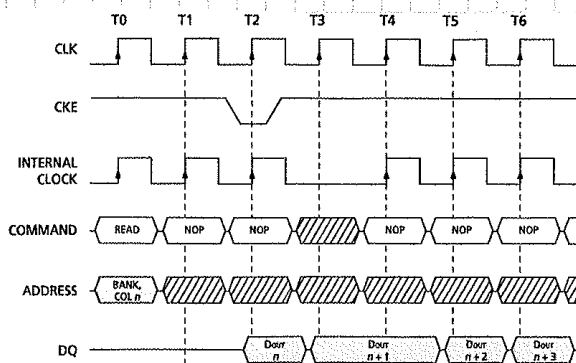
I segnali di ingresso rappresentano una codifica su 4 bit e rappresenta uno stato del sistema. Quindi, a seconda del CS, del RAS ecc ecc si hanno degli stati differenti:
 (DQM: output enable)



NOTE: For this example, burst length = 4 or greater, and DM is LOW.

Figure 22

Clock Suspend During WRITE Burst



NOTE: For this example, CAS latency = 2, burst length = 4 or greater, and DQM is LOW.

▨ DON'T CARE

Figure 23

Clock Suspend During READ Burst

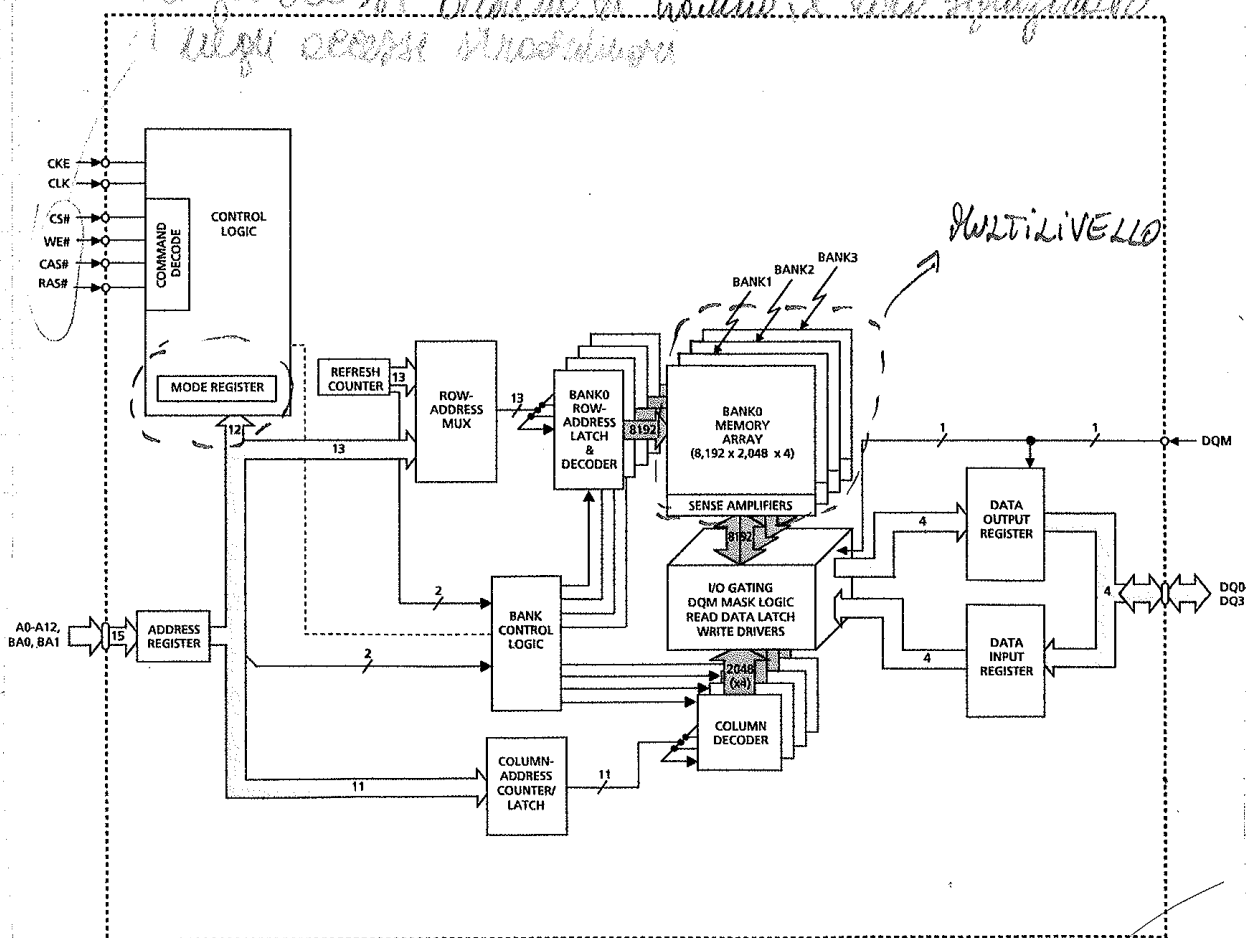


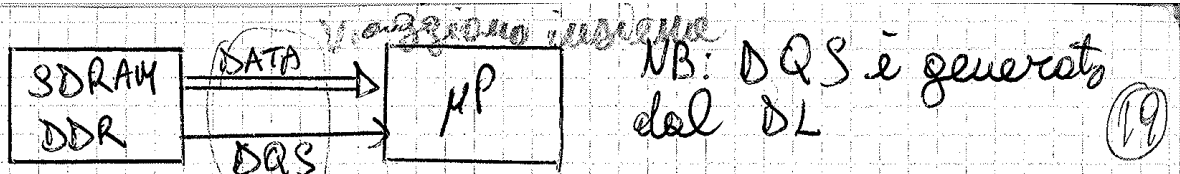
256Mb: x4, x8, x16 SDRAM

FUNCTIONAL BLOCK DIAGRAM

64 Meg x 4 SDRAM

*unfi stessi ordinari hanno il loro significato
 unfi stessi microprocessori*

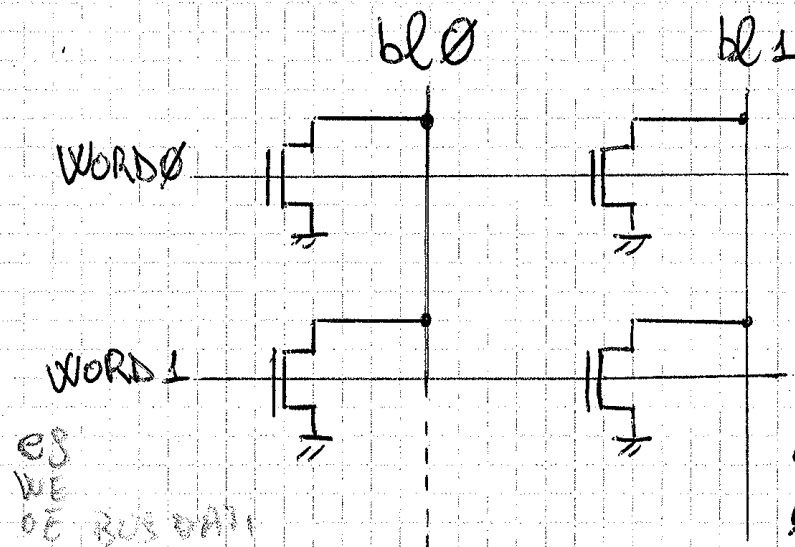




Quando arrivo al ricevitore, il DQS può essere utilizzato come il clock (perché in fase di progetto DQS viene fatto mandare insieme ai dati!!)

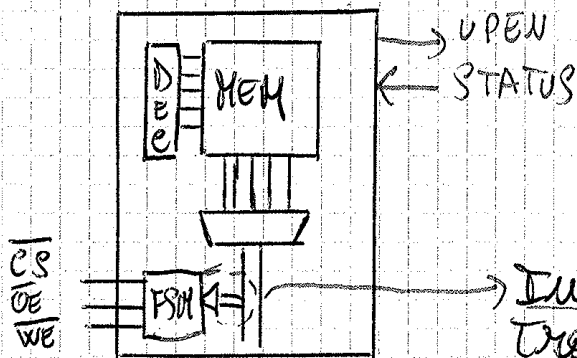
FLASH SERIALE: montato sulle DDR, al suo interno sono presenti tutte le varie configurazioni per poter programmare il controller della RAM

22/10/2013 Le DDR3 contengono il modo register, così come per esempio la fine dell' comando burst



MEMORIA FLASH di TIPO NOR
 NOR FLASH
 perché abbiamo una config wired-or: è sufficiente che uno dei transistor conduca affinché la linea veda a 0 logico (NOR)

Tali memorie sono indistinguibili dalla RAM Statica (i segnali di controllo sono gli stessi). Per quanto riguarda la lettura è IDENTICA alla RAM. (Si leggono le celle dell'ARRAY)
 In scrittura, non è necessario solo l'OE e il WE. Si distingue la scrittura sul BUS e sull'array.



Per sapere quando la scrittura è terminata vedere il bit di stato

In scrittura i comandi sono trasmessi all'unità di controllo!!
 quando legge, legge dall'array!

Scrittura flash

(22)

Nella flash la scrittura il tempo che ci vuole è lungo e indeterminabile, devo distinguere di operazione di scrittura dentro l'array dall'op. di scrittura sul bus (~~che è~~ ^{nell'interfaccia} il timing è uguale a quello delle RAM Statiche) quando si prende una SRAM se si legge si legge dall'array, se si scrive, si scrivono le celle della memoria; nella FLASH quando si legge si leggono le celle dell'array, quando si scrive NON si scrive nella memoria, ma si scrive nell'unità di controllo che c'è dentro



sol bus dati

la FSM è più sofisticata, quando si fa una lettura della memoria si legge dall'array, quando si fa una scrittura questi DATI NON VANNO nell'array ma sono dei comandi che vengono trasmessi all'unità di controllo c'è una asimmetria: quando leggo, leggo dall'array quando scrivo sul bus dati NON SCRIVO NELL'ARRAY ma scrivo dei comandi alla macchina e stati, nel timing quando faccio lettura e scrittura è identico (ex. ~~leggo~~ leggo in 50ms e scrivo sul BUS in 50ms, però ATTENZIONE, non ho fatto una scrittura dentro l'array, per scrivere dentro l'array magari ci vogliono 2ms) se io voglio cancellare un blocco della flash

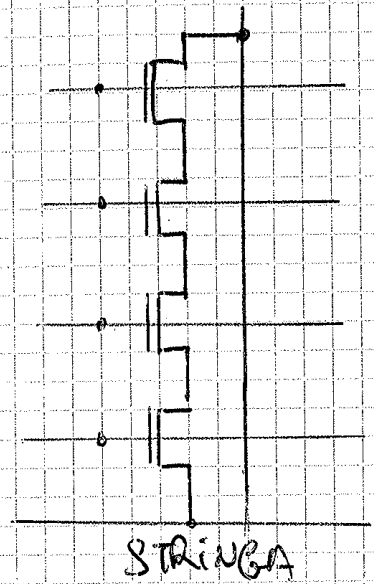
COSA C'È ALL'INTERNO COMMON FLASH INTERFACE?

ASPI 516759

I primi 3 byte della tabella sono QRY. (serve per identificare le [?] valori della tensione di alimentazione)

INTRODUCTION TO FLASH MEMORY

MEMORIE NAND: non ci si può accedere ad accesso casuale



Per un problema di computerizzazione è stata creata la NAND FLASH

F (larghezza di una diffusione)
Nelle NAND, dato che tutte le celle sono in serie ho bisogno di un solo contatto attraverso alla bit line. e sono più dense con meno cont.

non le singole word

Nelle NAND si legge la pagina (va molto bene per memorizzare ~~dei dati~~ ~~dei dati~~ ~~dei dati~~ dei dati sequenziali). Non posso usarle per memorizzare del codice (ci sono dei salti). Per ridurre la probabilità di errore si aggiungono dei bit (ridondanza) per poter risolvere dei problemi. per evitare che si scrive qualcosa in un blocco se quello è tutto scritto → WEAR OUT. invecchiamento di un blocco

WEAR LEVELING → tutti i blocchi invecchiano allo stesso modo

INTERFACCIA DI TIPO MULTIPLEXATA → per ridurre il numero

BUS a 8bit → (nelle applicazioni professionali di più ^{di più} ^(si chiamano) ^{8 bit alla volta})

NAND	NOR
	accesso casuale
	meno

NAND FLASH

(26)

Sempre gate flottante, consente una maggiore densità, hanno una probabilità d'errore alta. I dati sono raggruppati in blocchi che è la qto minima cancellabile. Sono memorie in cui l'accesso è prevalentemente ~~casuale~~ sequenziale.

Si ha una stringa di NOR messi in serie. La grosse differenza con le NOR è la compattezza (ci sono meno contatti).

Visto da fuori lo NAND assomiglia ad un disco, ^{come disco} cioè io non leggo una singola word ma leggo quella che viene chiamata pagina (che è l'unione di tutti gli oggetti sullo stesso string) quindi va bene per dati sequenziali. Non va bene per un singolo byte! Non va bene per il codice per i salti!

È dato che l'errore non è nullo si aggiungono dei bit di ridondanza (4) utilizzando codici di tipo polinomiale che mi permette in caso di singolo bit sbagliato di ricostruire l'informazione originale. Rispetto a un disco fisico ^{che} non si danneggia facendo lettura e scrittura nel caso di NOR e NAND ^{WEAR OUT} si ha un invecchiamento, le celle si danneggiano per ~~il~~ ^{gli} ~~dati~~ ^{dati} la struttura cristallina dell'ossido: non riesce più a scrivere! Si possono infatti avere delle tecniche come il WEAR LEVELING e invece di fare 1000 cicli di scrittura su un singolo blocco e nessuno sugli altri, cerco di fare invecchiare i blocchi allo stesso modo e miglioro le prestazioni complessive. Le NAND sono disp. in capote molto grandi e basso costo!

Il tempo di cancellazione le NAND flash è migliore (2ms VS 750ms) le NOR sono ancora utilizzate per il boot loader.

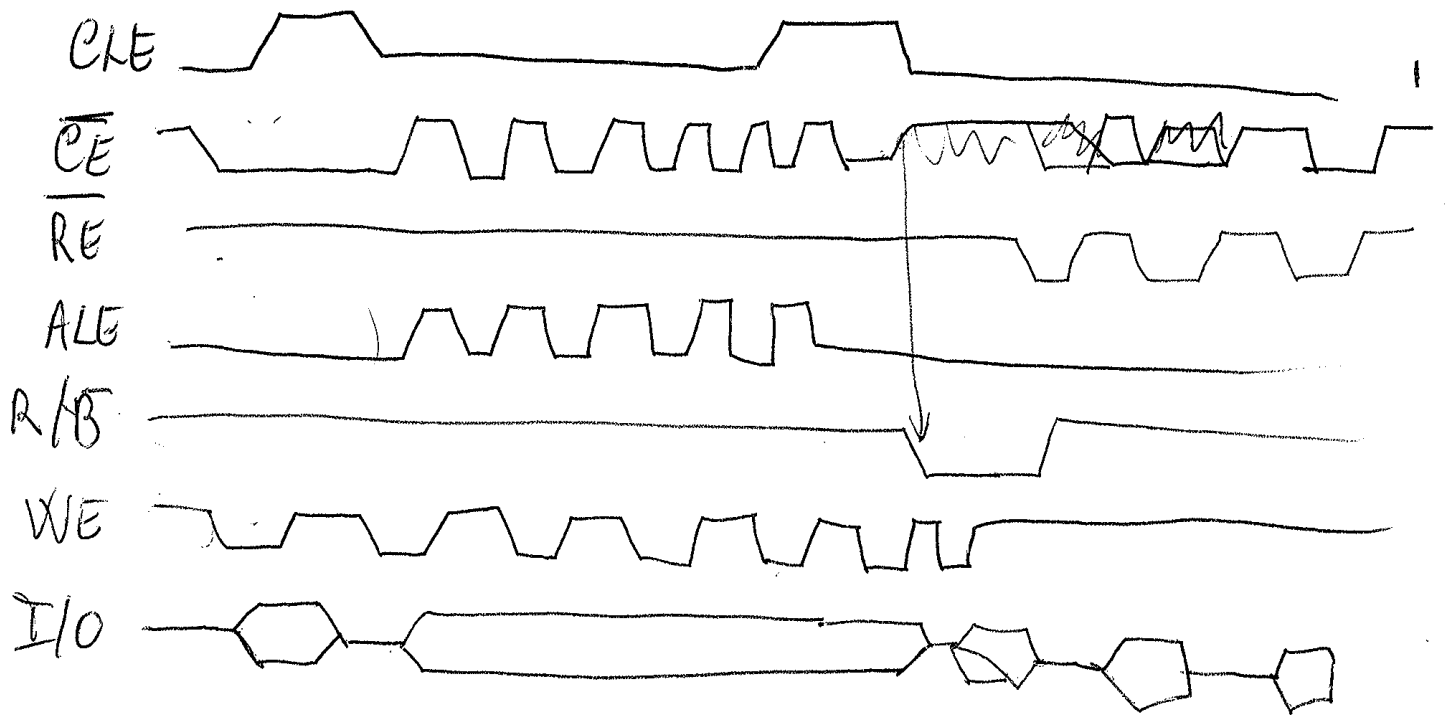
Per l'interfaccia ho 40 pin circa nelle NOR nelle NAND si hanno 21 fili di indirizzo (20 fili è 1 Mega, 21 vuol dire 2 Mbyte), la NAND in questione ne ha 23 pin: 5 pin di controllo & pin di I/O e poi write protect WP e ready/busy R/B ho lo stesso numero di pin indipendentemente dal numero di megabyte che ho dentro la flash, perché in questo protocollo, quando voglio leggere su dato io ~~non~~ mando un indirizzo in 5 operazioni (5 blocchi) quindi ho un indirizzo che è lungo 40 bit (5x8bit) e l'interfaccia ha sempre un numero di pin molto ridotto. Il numero di pin è standard per la compatibilità in futuro. Si ha nelle NAND moderne più banche di memoria per poter avere l'interfaccia.

OPERAZIONI

mentre nelle memorie NOR si ha una struttura simile alla RAM statica nelle NAND per migliorare le prestazioni le operazioni sono basate sulla pagina: quando do il comando di lettura io seleziono una riga intera (come nelle memorie normali) e lo carico dentro uno shift register (da 2000 byte) dopodiché la lettura avviene sequenzialmente e intanto l'array non sta facendo nulla, è vantaggioso come sistema.

Se voglio fare una programmazione faccio: carico nello shift register tutti i dati, ~~poi~~ trasferisco i dati

Quando faccio p di RW prima tiro fuori la pagina lo copio dentro lo shift register dopo ⁽³⁰⁾ lo sr viene caricato in fase di scrittura o viene scaricato serialmente in caso di lettura. La freq. di funzionamento è circa 33 MHz nel caso di lettura.



Nelle NAND flash si dividono in due piani fisici, si possono effettuare più operazioni contemporaneamente:

(32)

- leggere due pagine per lettore
- cancellare due blocchi;
- programmare 2 pagine.

Quando si hanno tanti bit bisogna anche avere un dispositivo performante. Si devono evolvere i "colli di bottiglia".

Oggi giorno nelle NAND il tempo di trasferimento fisico all'interno della memoria per i vari shift-register:

• SLC: 20-25 μ s MAX

• MLC: 50 μ s MAX

Il tempo di I/O:

- 2K page: 42 μ s

• 4K page: 86 μ s

Però l'interfaccia della NAND Flash è seriale, il collo di bottiglia è l'interfaccia che ci mette solo 40 MB/s (25 μ s)

In lettura c'è una penalty ma non in scrittura.

È stato introdotto uno standard chiamato "ONFI" ("Open NAND Flash Interface"), tutto ciò per standardizzare l'interfaccia NAND Flash. Senza questo standard le NAND FLASH avevano delle incoerenze tra di loro.

Permette di fare lettura e scrittura a 20 ns.

Nell'ONFI 2.0 si è cercato di avere una compatibilità anche con i dispositivi precedenti.

Si vuole un dispositivo "scalabile" aggiungendo il segnale di strobe DQS, aumentando il n° di piani di NAND.

Le NAND HIGH-SPEED, sono quindi compatibili con le STANDARD.

29/10/2013

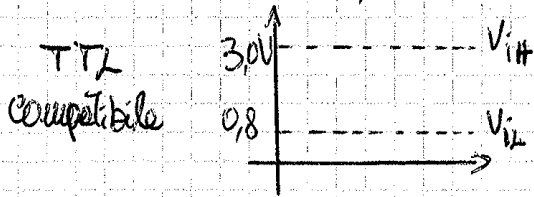
~~LEZIONE CHE SPIEGA IL LAB 11~~

(34)

I) Fisico (voltage levels)

Eg. 3,3V CMOS

LEZIONE CHE SPIEGA IL LAB 11
NON UTILE AI FINI DELL'ESAME
(pag. 34-35)

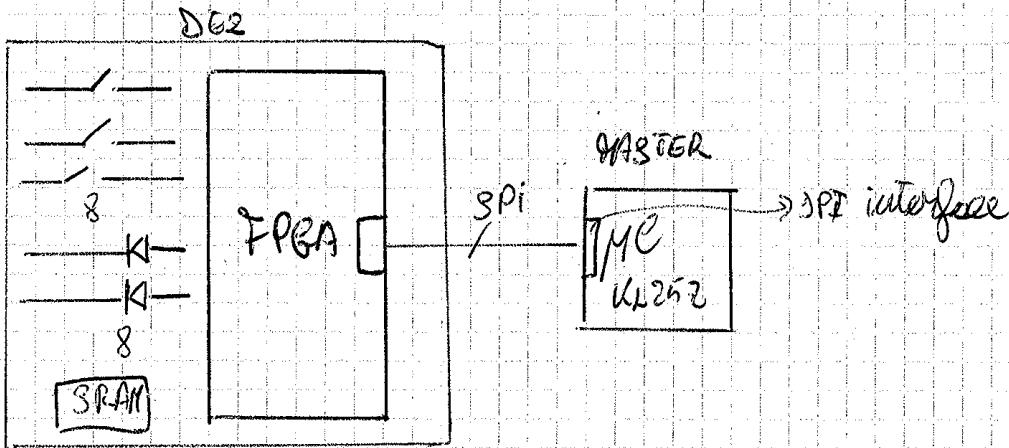


Primo livello di specifiche di interfaccia

II) Protocollo - low level

per esempi I²C, SPI, Parallelo

III) Protocollo - high level



Si possono identificare 3 registri: controllo, stato, dato.
Reg di controllo: reg che contiene dei bit i quali modificano il comportamento delle periferiche (per esempio: spegnimento delle periferiche).

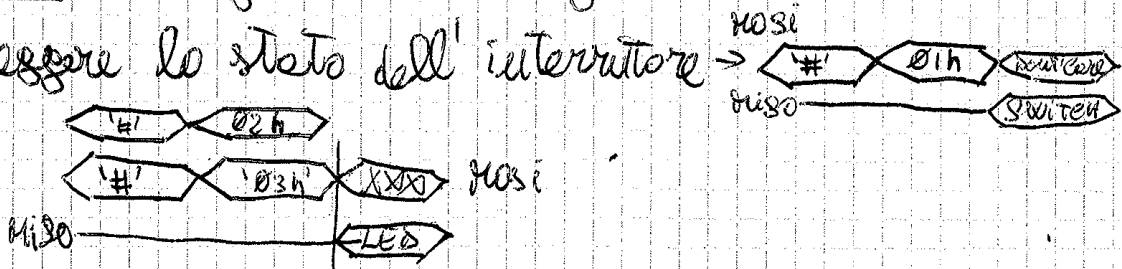
Reg di stato: controlla lo stato di una periferica.

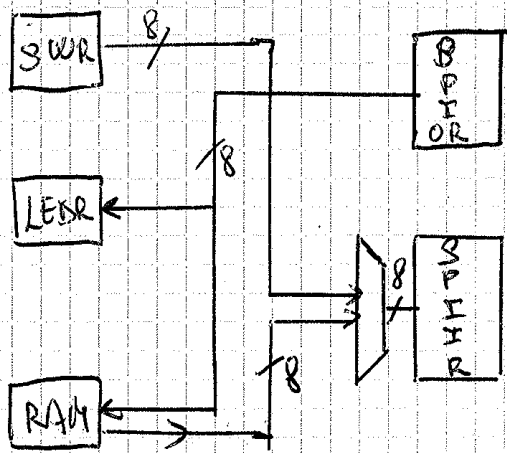
Reg di dato: contengono un dato generico

Si vuole leggere lo stato dell'interruttore

Write led

Read led



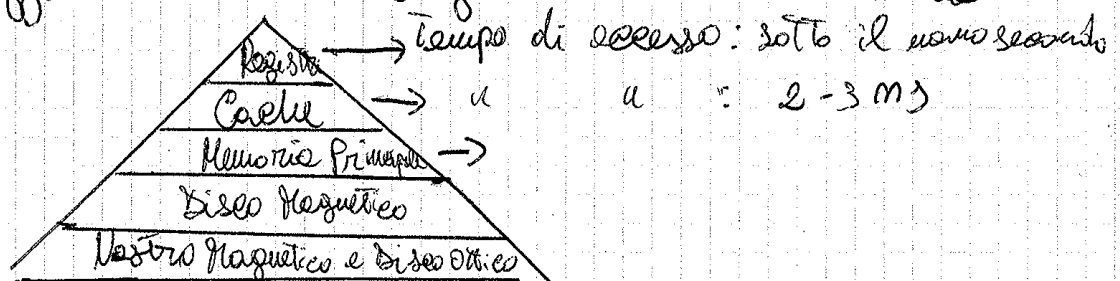


(36)

11/11/2013

MEMORIA IN UN SISTEMA DI ELABORAZIONE

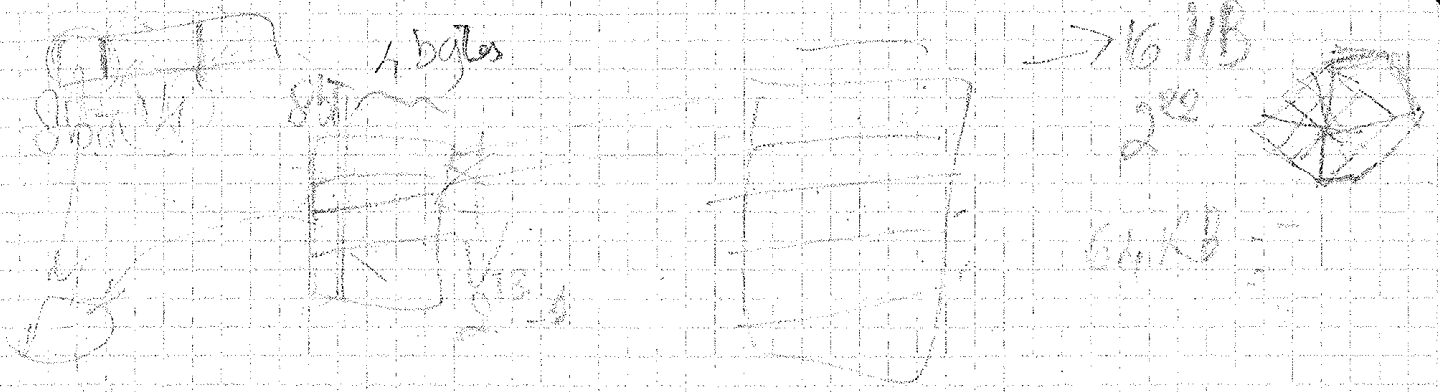
Le memorie dovrebbero essere attaccate al μP , ma quello che succede è che le memorie hanno un tempo di accesso troppo grande per il μP e l'obiettivo sarebbe che i tempi siano uguali tra mem e μP . Il microprocessore lavora quindi ad una velocità molto più alta della memoria. Una memoria più è piccola più è veloce, quindi una possibile soluzione è metterne più di una affiancate. Ho una gerarchia di memorie



Perché la gerarchia funziona bene devo avere i dati che mi servono più vicini alla cima della piramide. Devo introdurre un livello di estrazione che mostri al μP la gerarchia delle memorie. Una **CACHE** è una memoria piccola che contiene i dati della memoria principale e le istruzioni che servono al processore. Poiché il proc. crede sempre di essere attaccato alla memoria vi è il **CACHE CONTROLLER** che serve il μP nelle richieste dei dati. Se il dato non si trova in CACHE si accede allora alla mem. principale. Si definisce **HIT** l'accesso ad un dato presente nella CACHE e **MISS** quello mancante.

$$\text{HIT RATIO} = \frac{n^{\circ} \text{ hit}}{n^{\circ} \text{ accessi}}$$

$$\text{MISS RATIO} = \frac{n^{\circ} \text{ Miss}}{n^{\circ} \text{ accessi}}$$



- Vantaggio: semplice ed economico, il tag è costo
- Svantaggio: se accedo a 2 blocchi essi si contengono in uno spazio della cache. Per ovviare a questo problema si ha un: "TWO WAY ASSOCIATIVE SET" (pag. 5/11). Si ha un miglioramento di hit ratio. In questo modo ho un TAG più grande. Quando faccio un accesso adesso devo fare un confronto su 2 linee anziché solo in una. Devo essere un bit di VALID per (sentire registrazione) (ora 18:00).
- FULLY ASSOCIATIVE MAPPING (non usato perché molto dispendiosi) - esempio di utilizzo delle CAM.

Approfondire memorie virtuali da slide fornite da Ryo Roch

e li colloco su un disco grande, lo spazio di indirizz. lo divido in pagine (4KB) e tutte le volte che il proc. ha bisogno di un'informazione prendo la pagina la carico nella mem. principale e poi accedo lì. ~~Quando~~ Quando sono nella memoria principale queste pagine si chiamano frame.

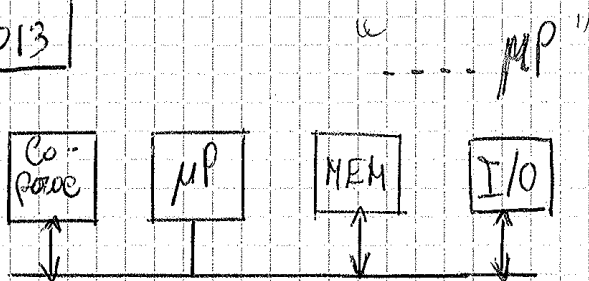
In memoria tengo le pagine che utilizzo.

Quando la CPU chiede di accedere a una locazione virtuale che non è già nella memoria principale se dice che ho un page fault (ho cercato di accedere a una pagina che non è in memoria ma sul disco).

L'indirizzo virtuale contiene il numero della pagina
l'indirizzo fisico contiene il frame e l'hardware controlla se c'è o meno

12/11/2013

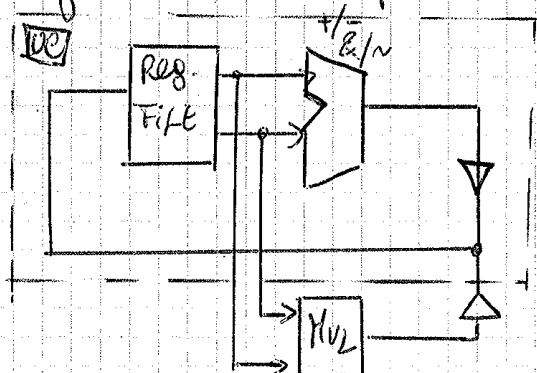
(42)



Se voglio migliorare il μP inserendo delle funzioni specifiche o anche modificare l'EK avendo costi maggiori inserisco un co-processor che espande le istruzioni.

Collego il co-processor attaccando direttamente una porta avere over-head di comunicazione. Il co-processor può però compiere in molti meno colpi di CLK delle operazioni che il μP impiegherebbe troppo tempo ad eseguire e quindi si ha in ogni caso un guadagno.

Si può collegare il co-proc. in maniera più intima?



→ ha un unite di controllo, più dei busi dati, istruzioni che non hanno una sequenza

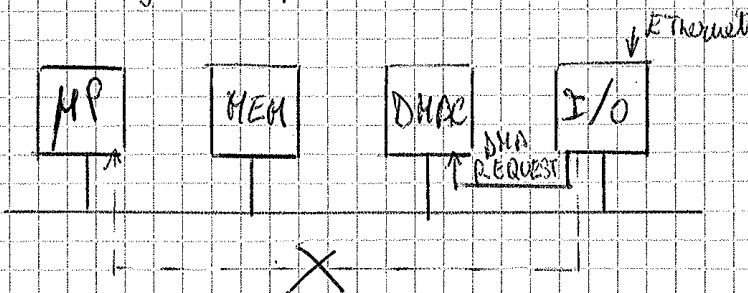
In questo modo non ho over-head di com., sarebbe corretto parlare di un proc. potenziato (augmented-processor)
Come faccio a dire al programmatore che esiste anche il co-processor?

Scrivo un programma solo che gestisce anche il co-proc. Questo lo si fa in maniera tale che il μP prevede dei "BANCHI"

Un SW non è in grado di riconoscere se è avvenuto un interrupt. Il proc. vede l'interrupt, lo esegue e ritorna a lavorare, quest'operazione si chiama trap. Se non c'è il co-proc. se c'è un trap il codice fa le operazioni "manu" se esiste il coproc in un tempo minimale si eseguono le operazioni

DMAE - Direct Memory Access Controller
 tecnica di fare operazioni di I/O con le memorie senza il μP

(43)



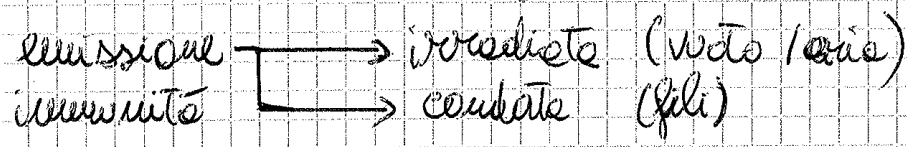
il DMA manda una richiesta alle periferiche e una direttamente alla RAM. Si genera un segnale di ACK (Acknowledge)

19/11/2013

"COMPATIBILITÀ ELETROMAGNETICA"

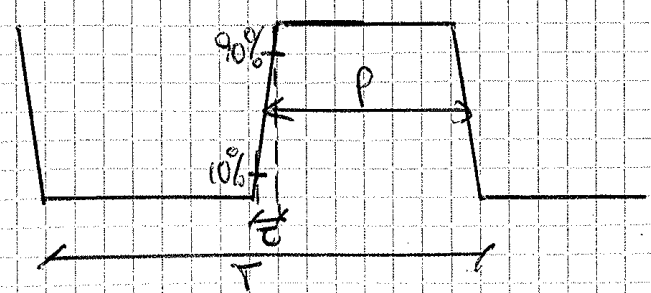
A livello nazionale le norme sono regolate da CEM, ~~internazionale~~ ^{comunità europea} il CENELEC e a livello mondiale le ISO. Non solo bisogna rispettare le norme a livello elettrico/meccanico ma anche quelle di compatibilità elettromagnetica.

Il sistema deve essere compatibile (non deve disturbare altre apparecchiature ^{mentre funziona} ed è sotto dei limiti; il ricevitore delle onde deve essere "robusto" e deve sopravvivere) con altre elettromagneticamente con l'esterno.



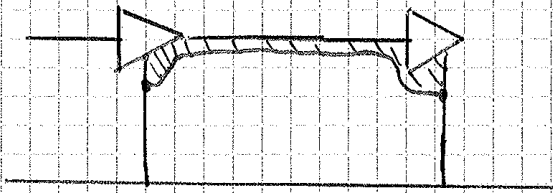
Per il teorema di reciprocità se progetta un'antenna che trasmetta poco essa riceve anche poco.

Nei segnali periodici si ha:



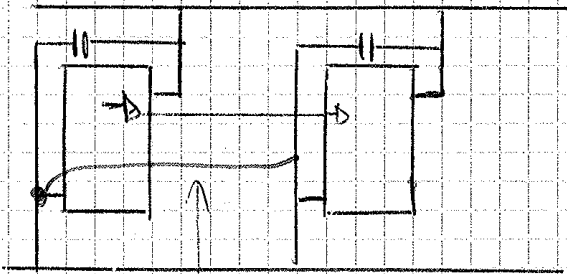
Se ne costruisce una maschera:

Le I_L si migliora aumentando il rise time, l'Area si riduce trovando un filo di massa che possa vicino l'altro filo (45)



La corrente passa dal filo più corto perché ha impedenza minore

La corrente percorre sempre il percorso che sottende area minima, poiché ~~è~~ è meno induttivo e quindi meno impedenza.



i disturbi vengono ridotti

MODO COMUNE

Pezzo di conduttore agitato da un'estremità



Nel caso di ~~modo~~ modo comune il campo irradiato è uguale a:

$$E = \frac{I_{cm} \cdot l}{R}$$

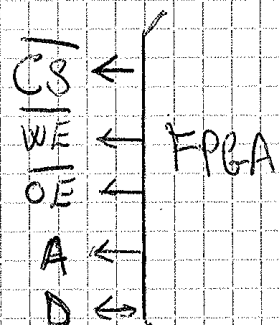
Moltiplicando il guadagno di antenna per l'avvolgimento del segnale si ha:

Si può aumentare l'indutanza per esempio mettendo una ferrita

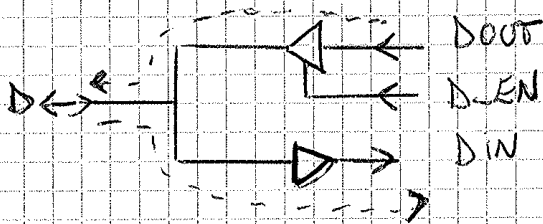
(47)

28/11/2013 ANCHE QUESTA LEZIONE (PAG 47-48) SERVE PER IL LAB NON PER L'ESAME

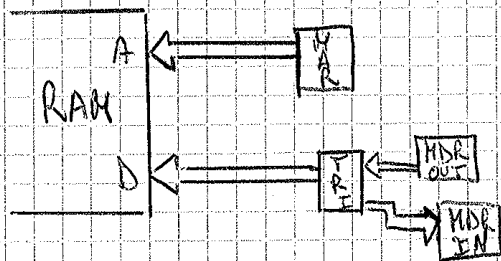
COME GESTIRE LA MEMORIA



Il ~~dato~~ segnale D è bidirezionale, si può leggere o scrivere. Per farlo diventare ^{unidirezionale} ~~bidirezionale~~ si fa così:



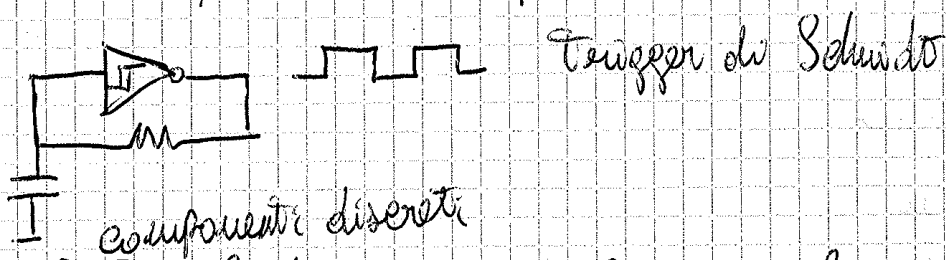
```
IF D_EN = '1';
  D <= D_OUT
ELSE
  D <= 'Z'
END IF
```



7/1/2014

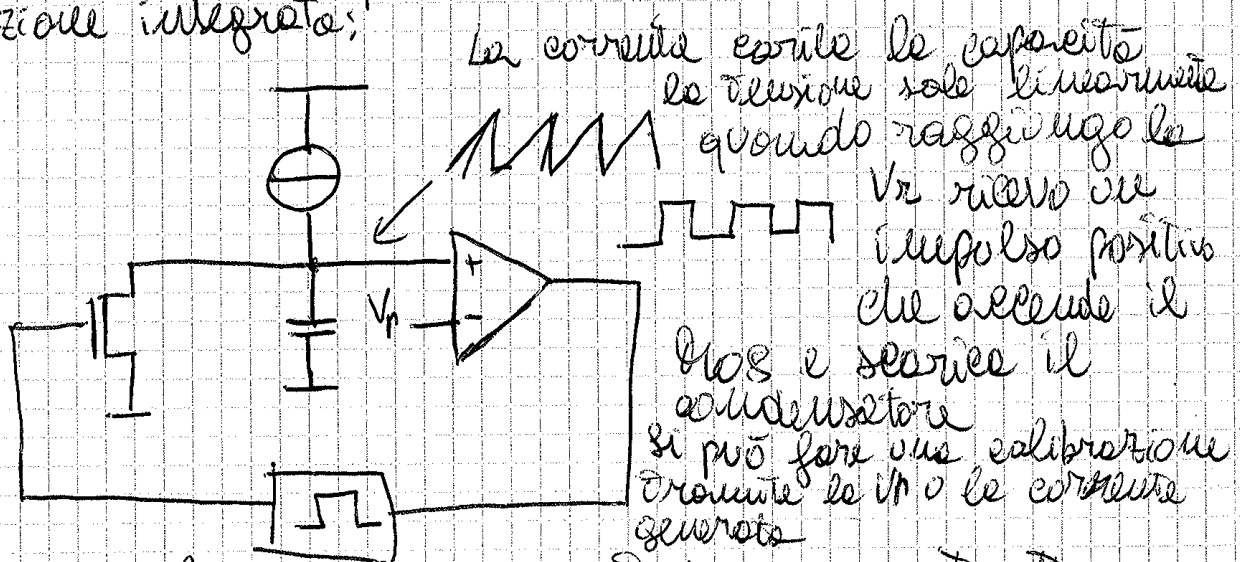
GENERARE SEGNALI DI CLK

Per generare il clock, come visto precedentemente:



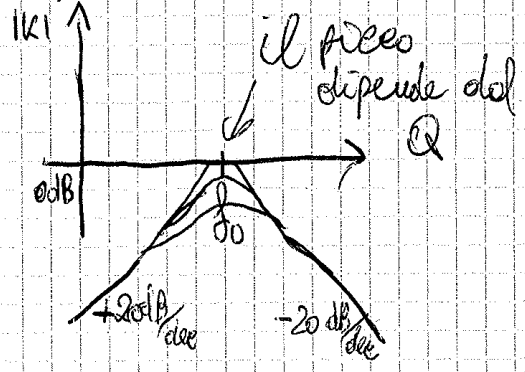
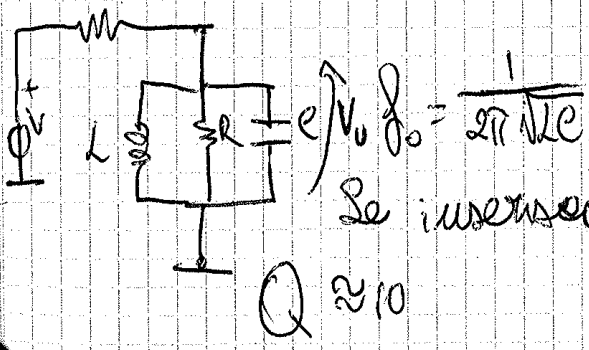
I componenti discreti
 Oggetto economico, ma il difetto sono le R e le C poiché si hanno delle tolleranze (dell'ordine del 10-15%) nella frequenza generata) e si ha anche un'altra 20% dovuta alle tolleranze per l'isteresi. Errore totale del 30-40%.

Se si vuole avere l'integrato è un problema poiché è difficile costruire i componenti precisi, se si vuole avere una soluzione integrata:



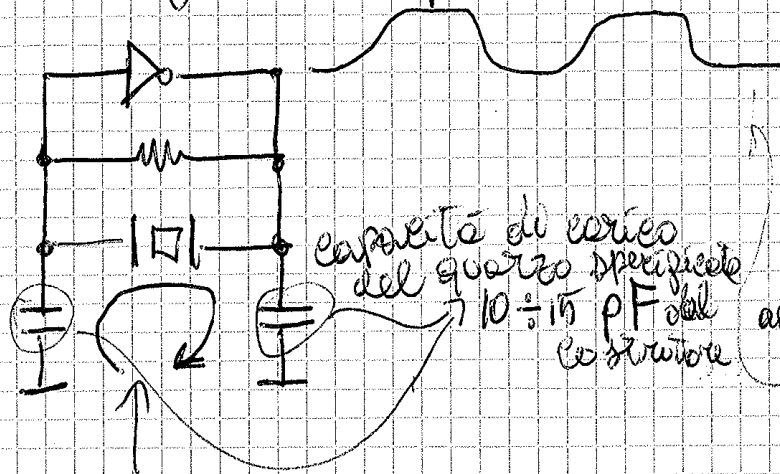
Oscillatore e rilassamento, sono tutti componenti. Attraverso il C che viene messo esternamente all'integrato perché i valori per stocci dentro devono essere precisi.

I μC a basso costo hanno oscillatori come questo. Per avere un oscillatore più preciso si vuole passare da un circuito RC ad un RLC risonatore. Un risonatore parallelo ha una curva di impedenza:



Se si vuole integrare il quoziente (si mette in retroazione) (51)

(questa pagina ha dei richiami a pagina 52 con i simboli \otimes | \odot | \triangle)

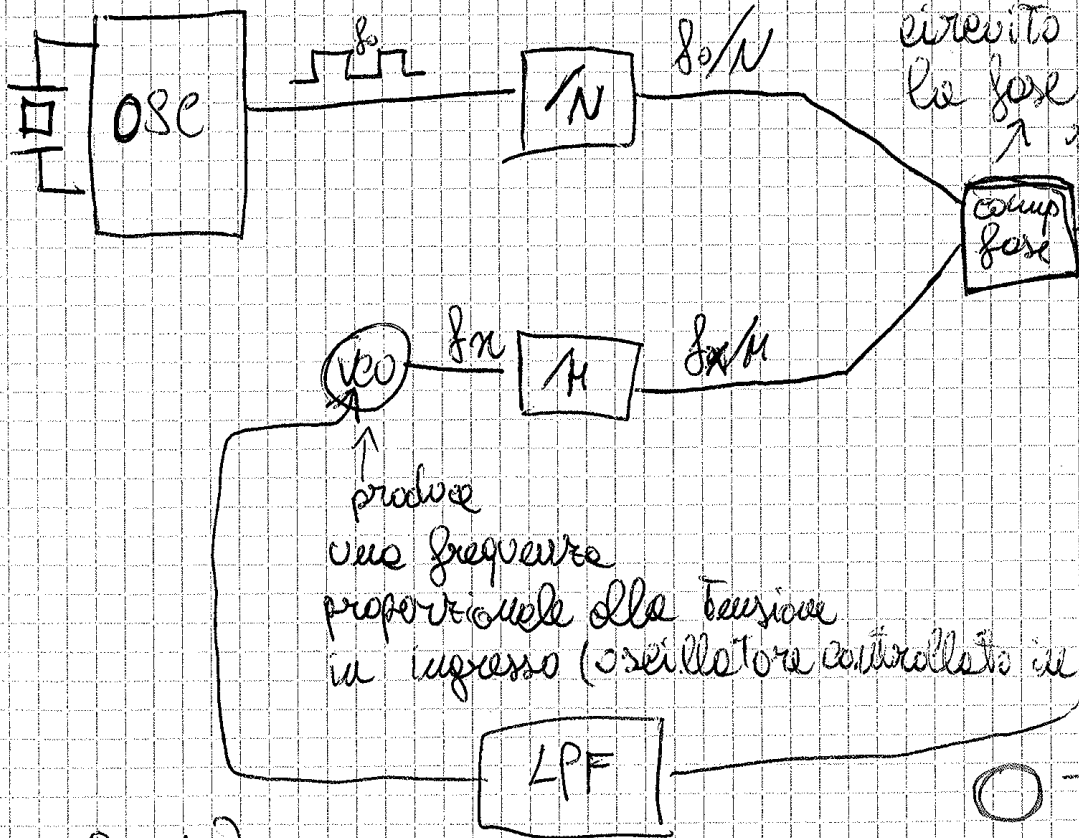


per avere l'onda corretta lo si inverte ad un altro invertitore

Circuito in cui posso corrente

L'amplificatore serve per ripristinare l'energia persa dall'oscillazione (per le perdite meccaniche).

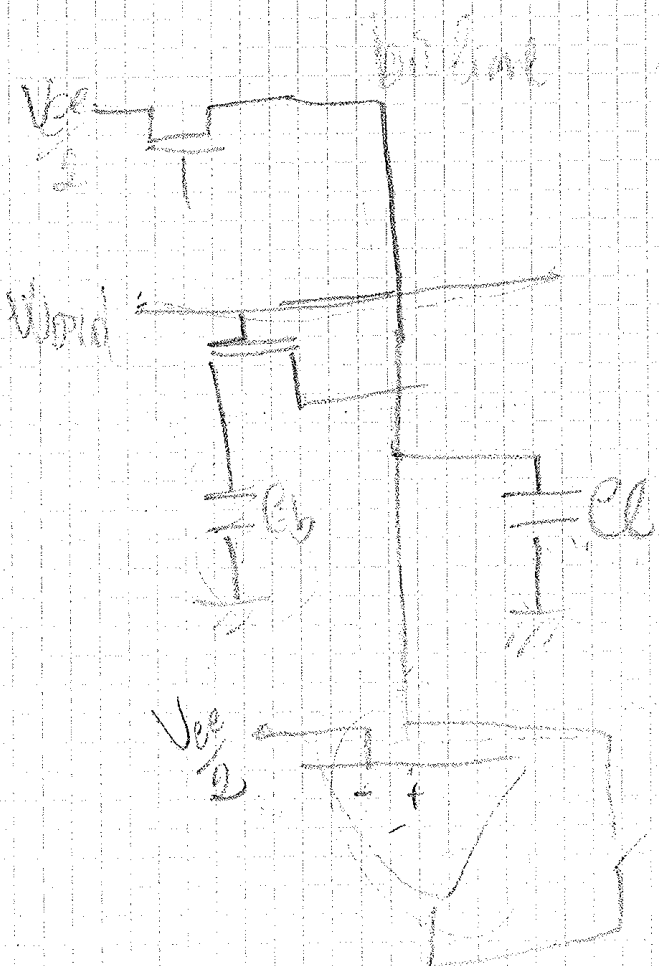
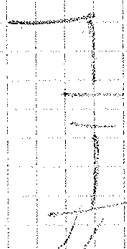
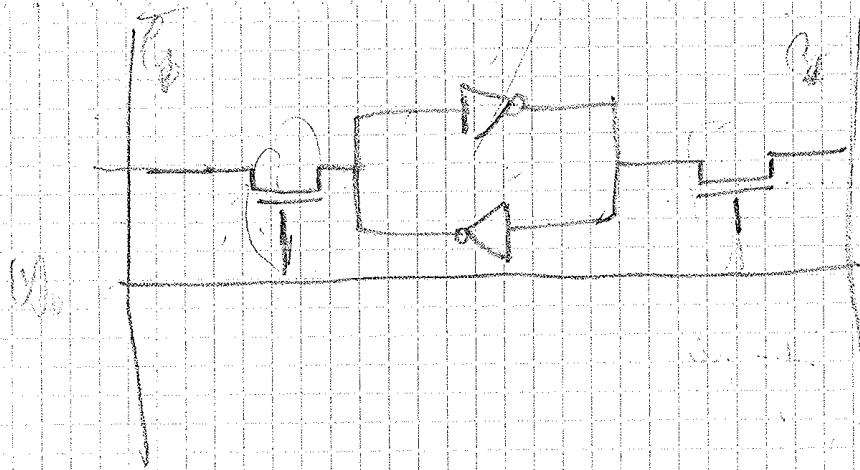
freq. di risonanza
 \otimes LF 32768 Hz
 \rightarrow HF 1 KHz \div 20 KHz



controllato
 un de segnale positivo se uno è in ritardo rispetto all'altro e negativo se il primo è in anticipo
 in tensione

$$\left. \begin{aligned} f_o/N &= f_x/M \\ f_o &= f_o \frac{M}{N} \end{aligned} \right\} \text{lo impone l'anello di retroazione}$$

Il quarzo ha una tolleranza chiamata tolleranza iniziale di $\pm 100 \text{ ppm} \div 10 \text{ ppm}$ specificata ad una certa temperatura ambiente



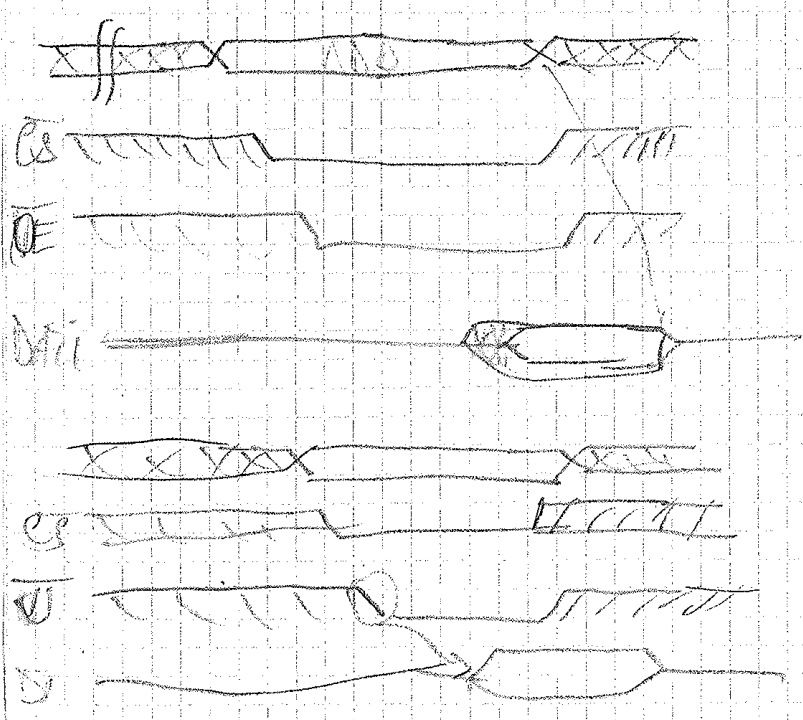
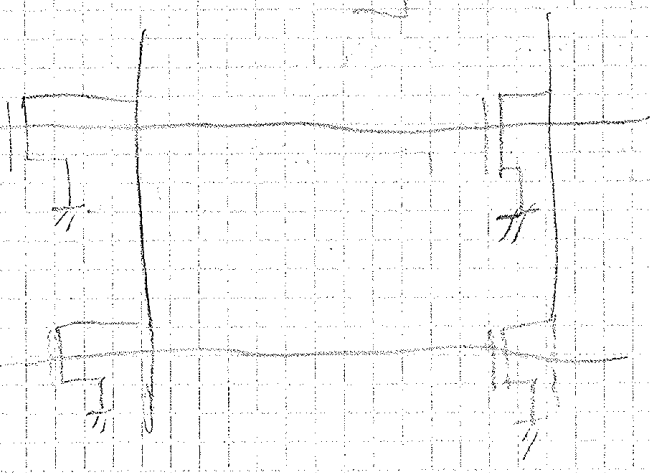
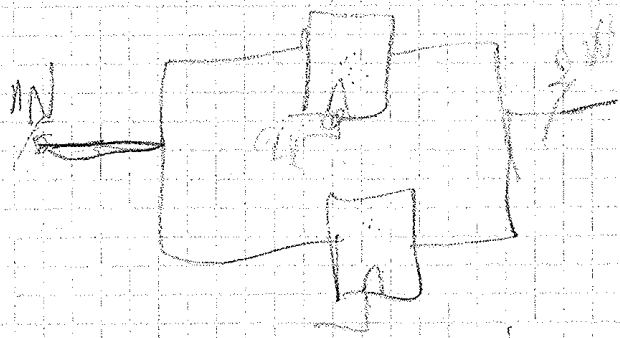
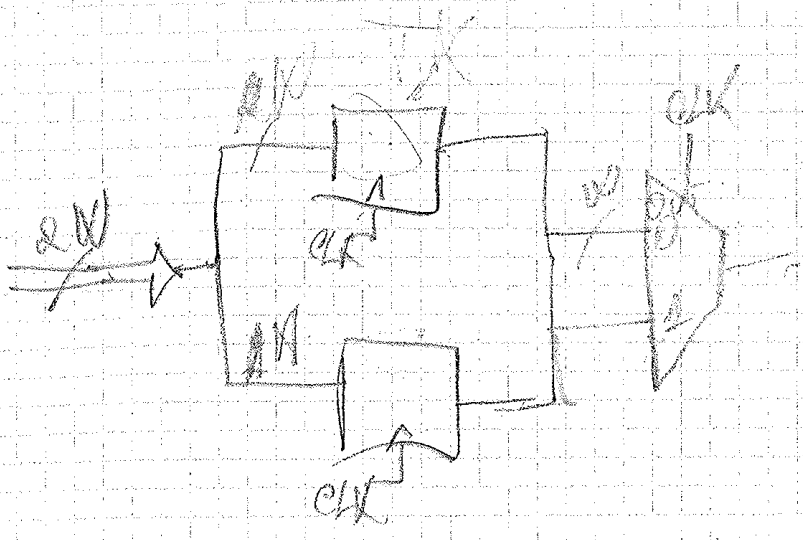
$$Q = CV$$

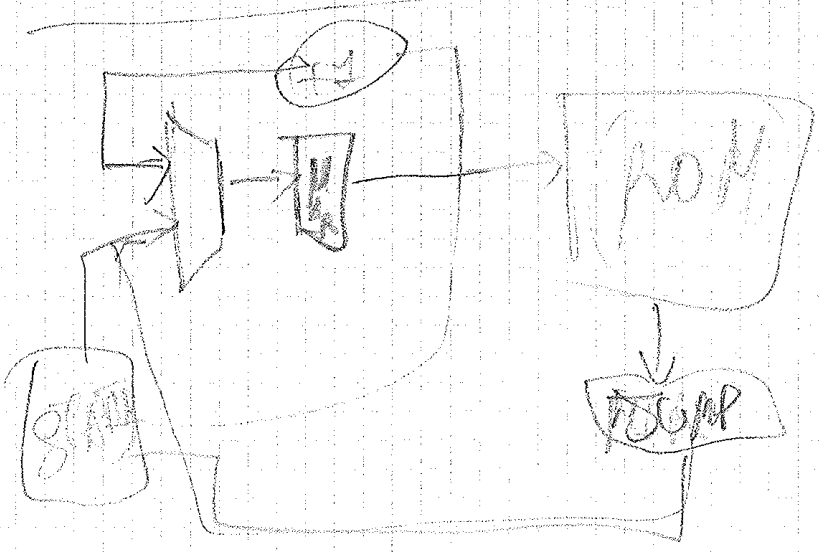
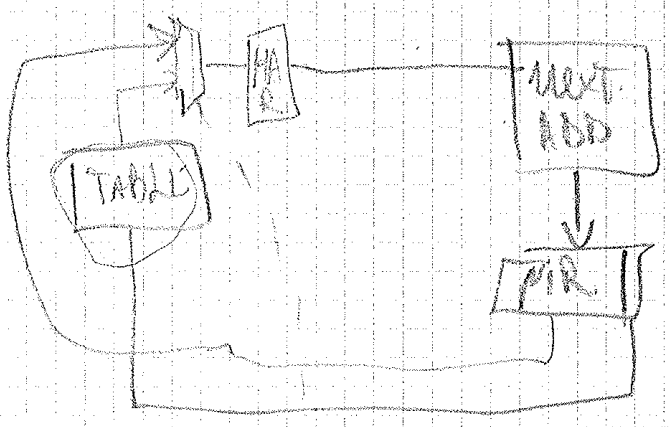
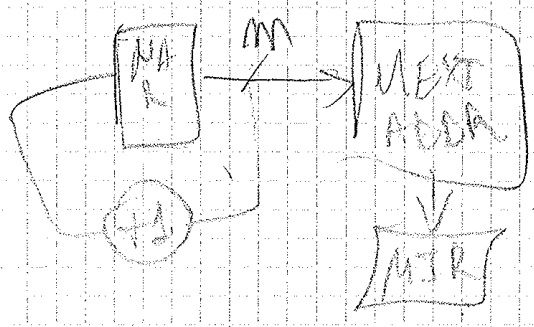
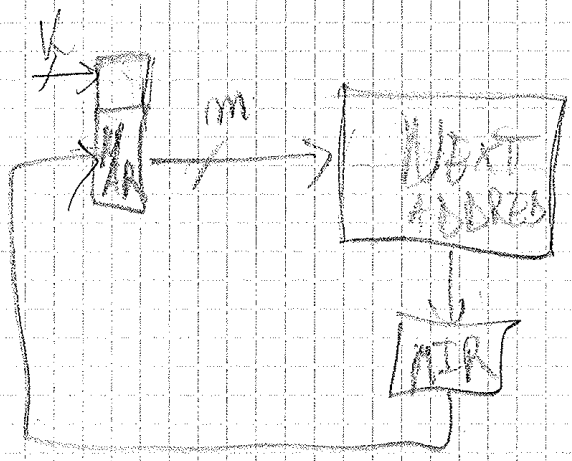
$$Q(t_0) = C_b V_{ce} + \frac{V_{ce}}{2} C_L$$

$$Q_{t=500} = V_{ce} (C_L + C_b)$$

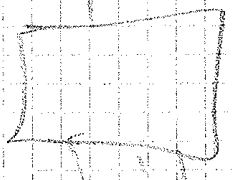
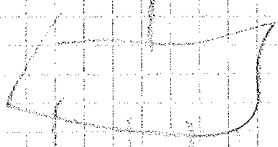
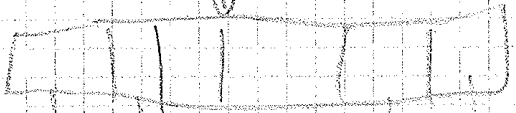
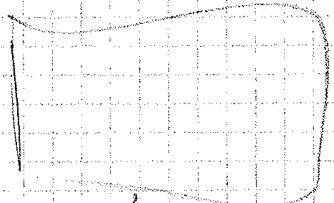
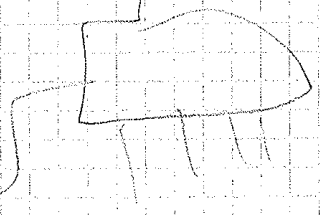
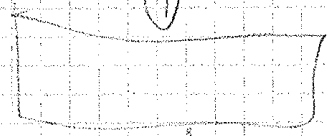
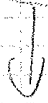
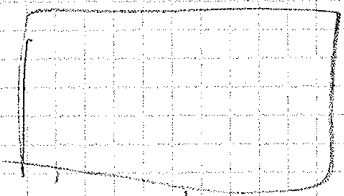
$$\frac{C_b V_{ce} + \frac{V_{ce}}{2} C_L}{C_L + C_b} = V_{out}$$

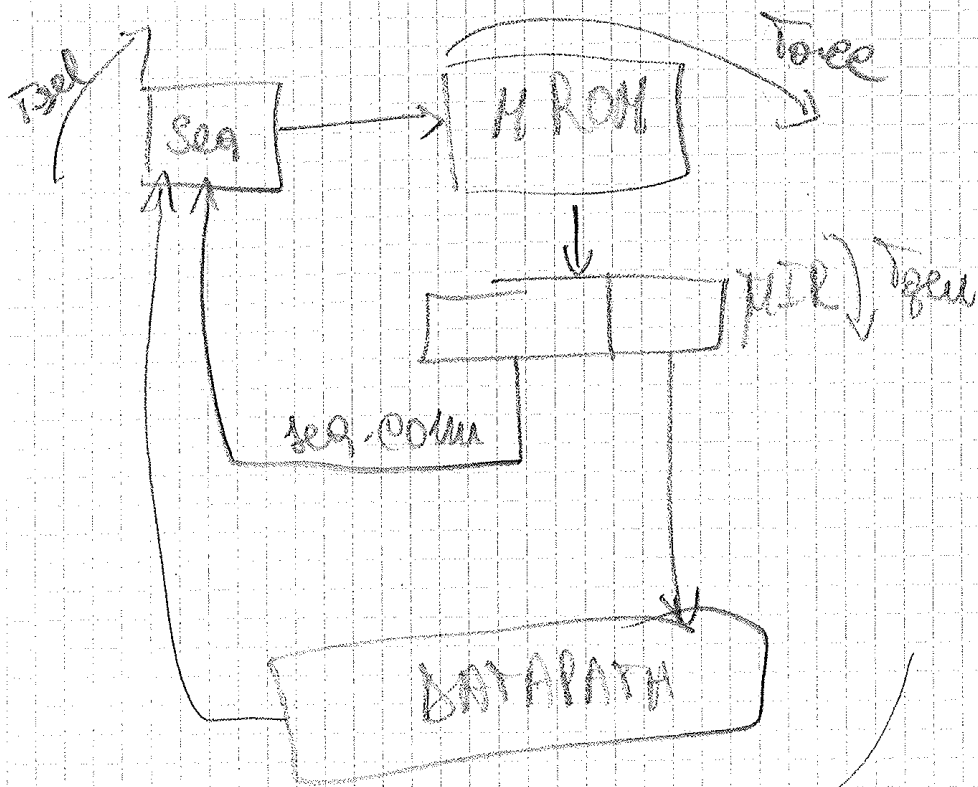
$$\frac{V_{ce} C_b}{C_L} + \frac{V_{ce}}{2} = V_{out}$$





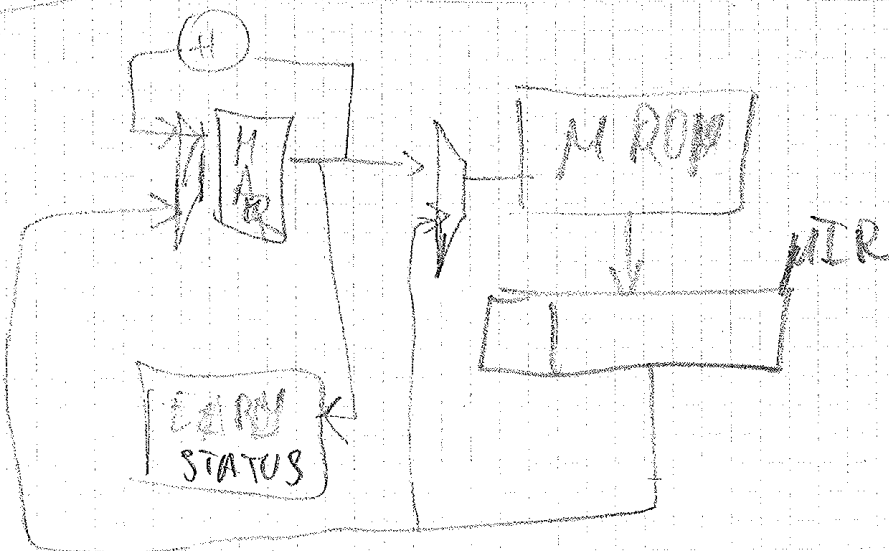
over end

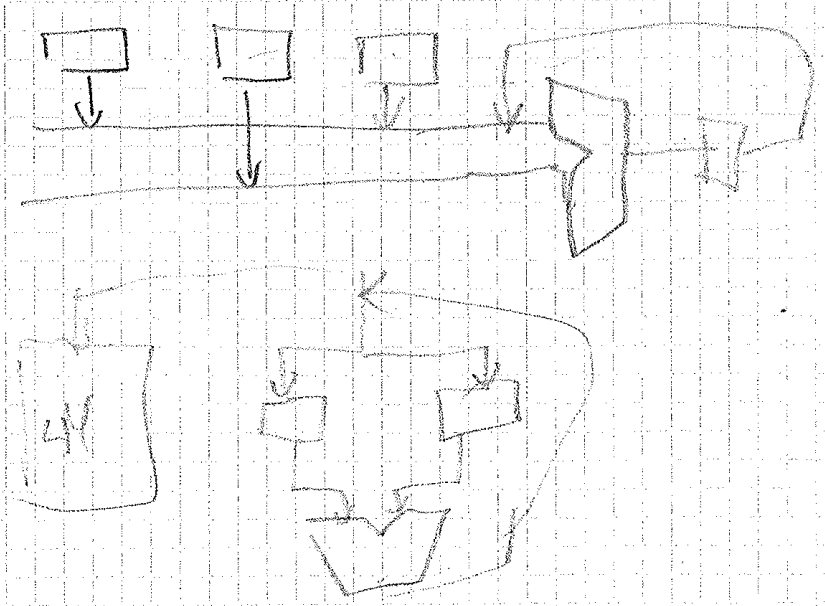
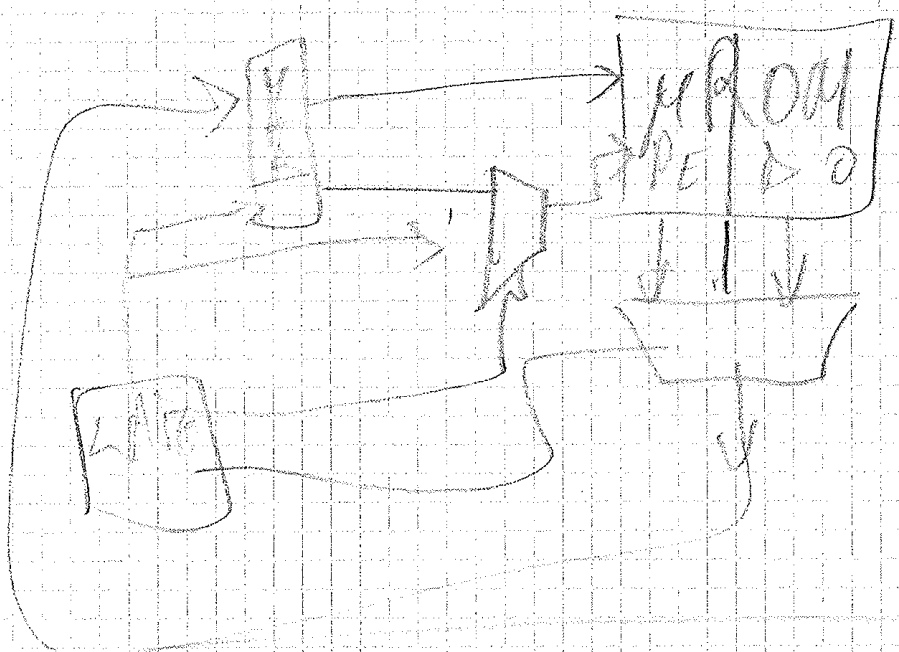
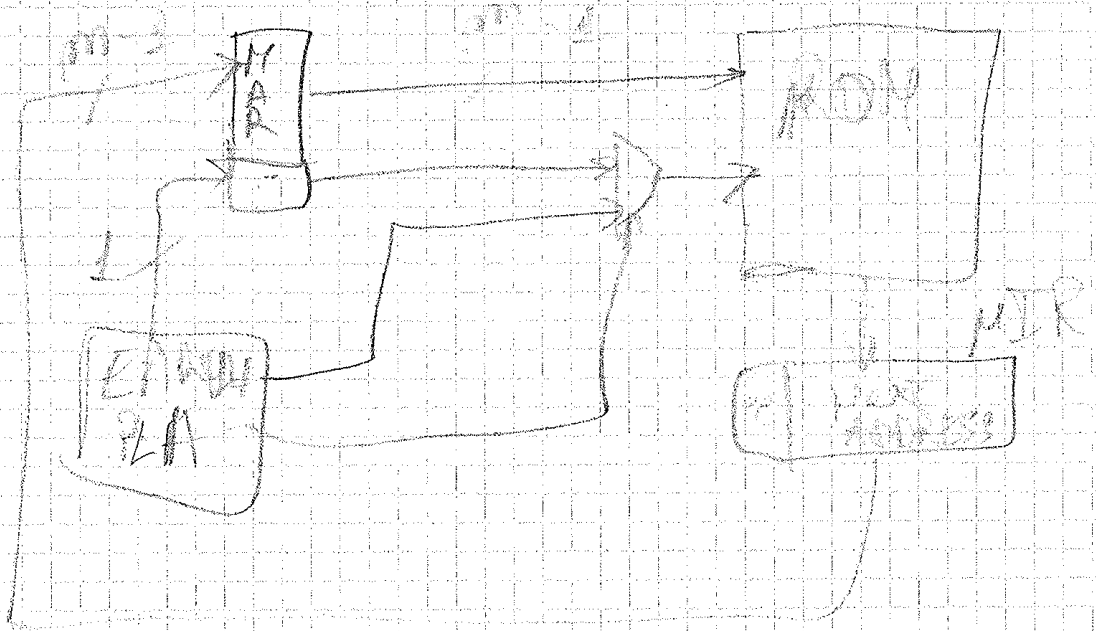


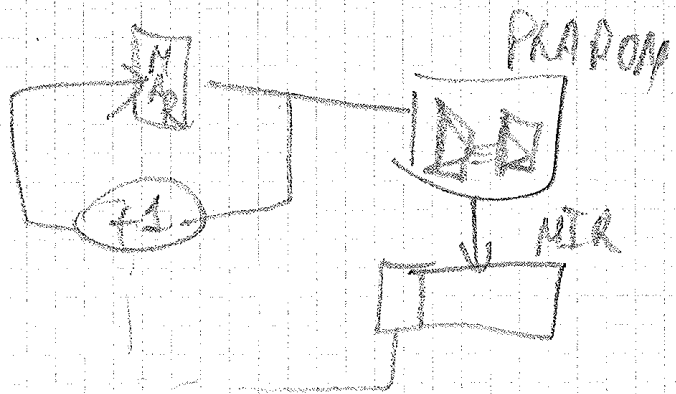
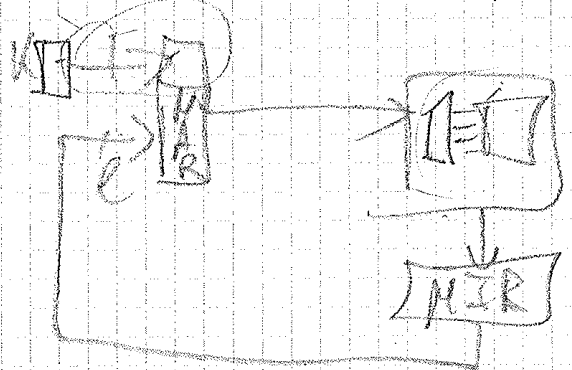
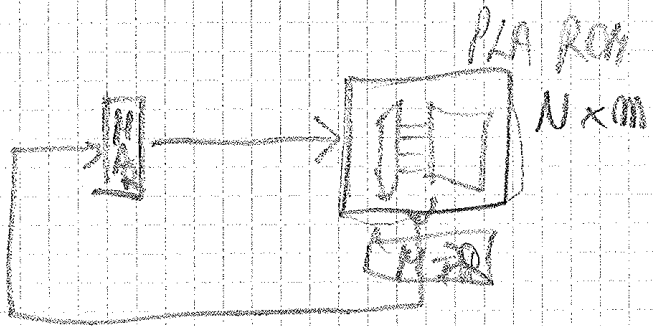
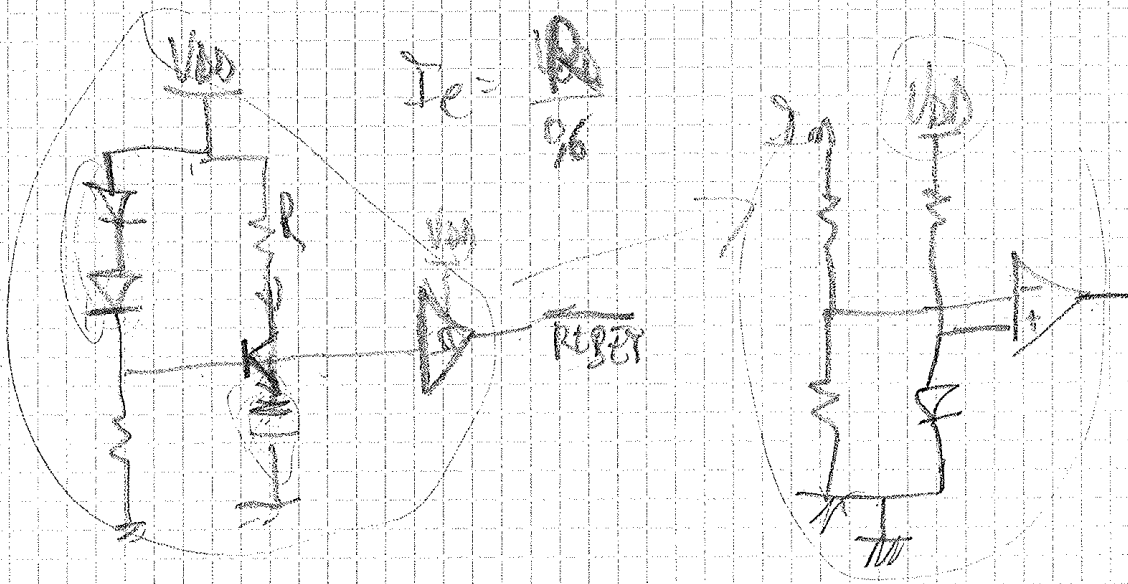


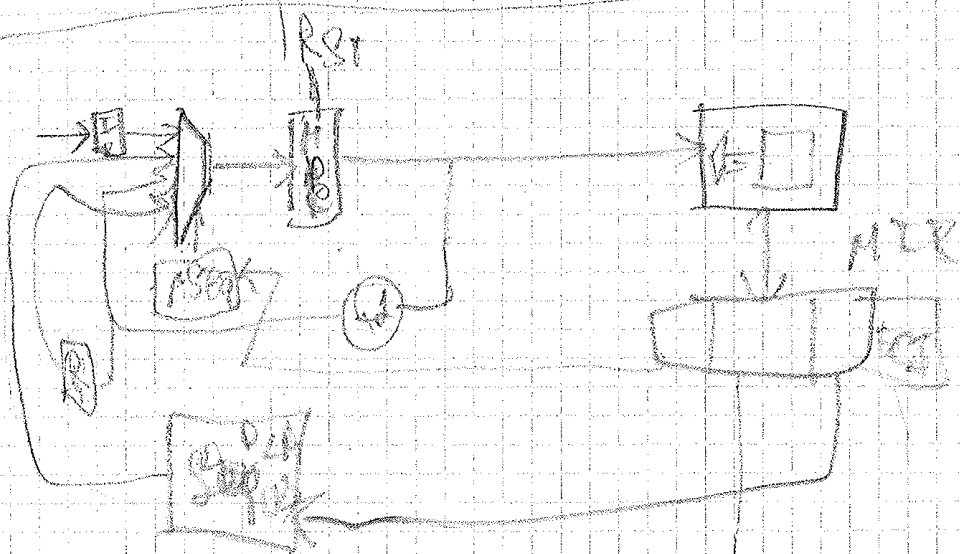
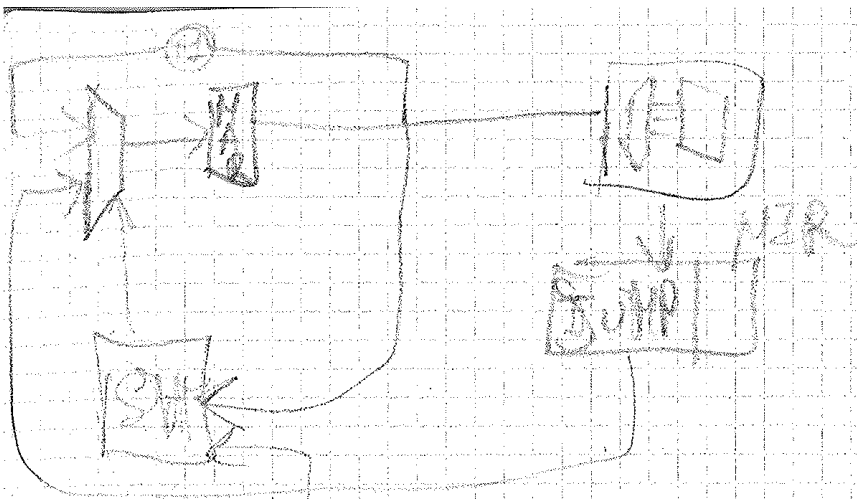
TOP

$$T = T_{sel} + T_{oree} + T_{gen} + T_{top}$$

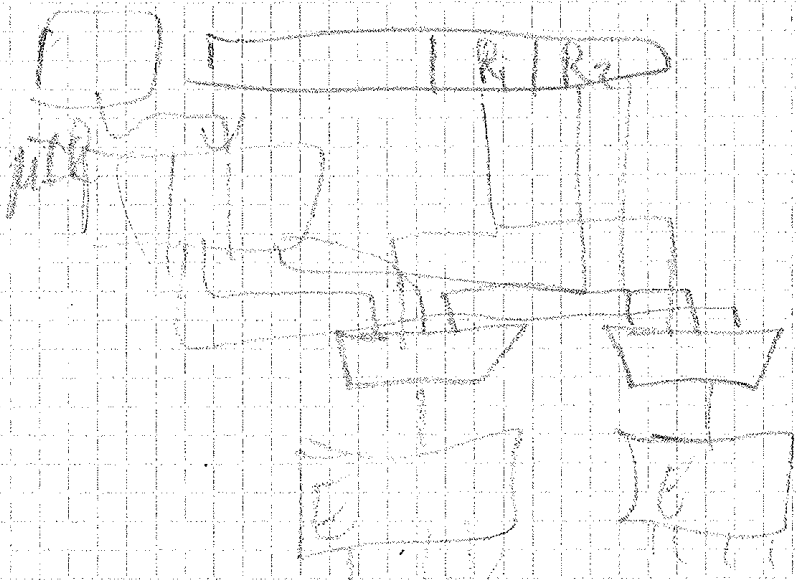
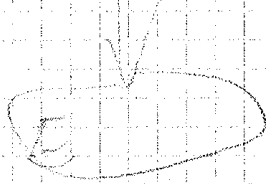






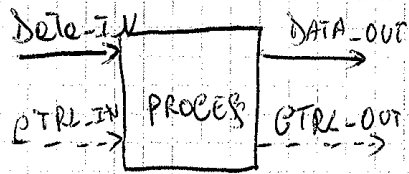


Handwritten text: $\frac{1}{R_1} + \frac{1}{R_2}$

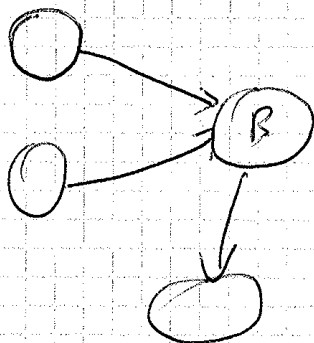


Parte Zamboni (04/10/2013)

Data-flow-diagramma di processamento dei dati, visione dei dati che deve elaborare l'algoritmo



dal DFD si costruisce la Control-DFD
il DFD fornisce cerchi e blocchi:

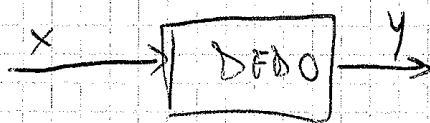


Gli archi forniscono i dati, i cerchi le operazioni da seguire
E un processo speciale numerato.

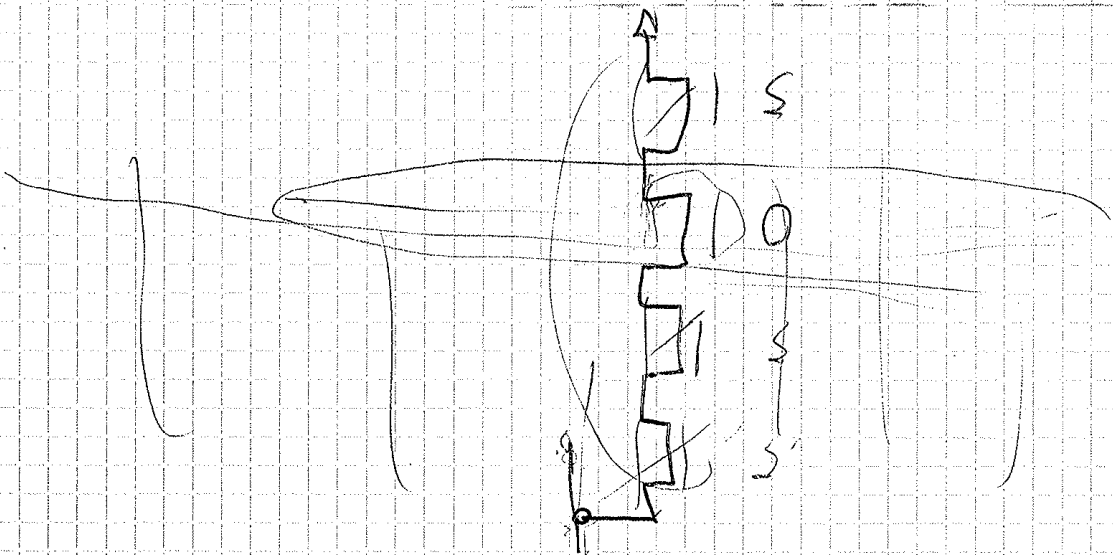
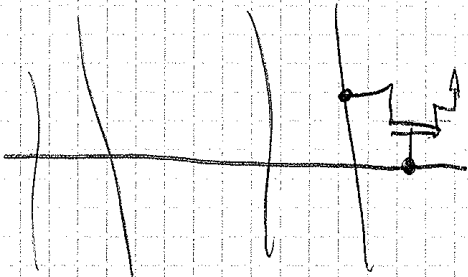
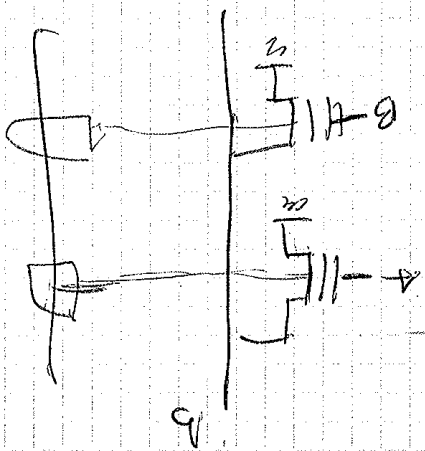
Esempio

Proiettore su filtro:

$$Y(m) = \sum_k a(k) X(m-k) \quad k=0 \dots 3$$



3474158448



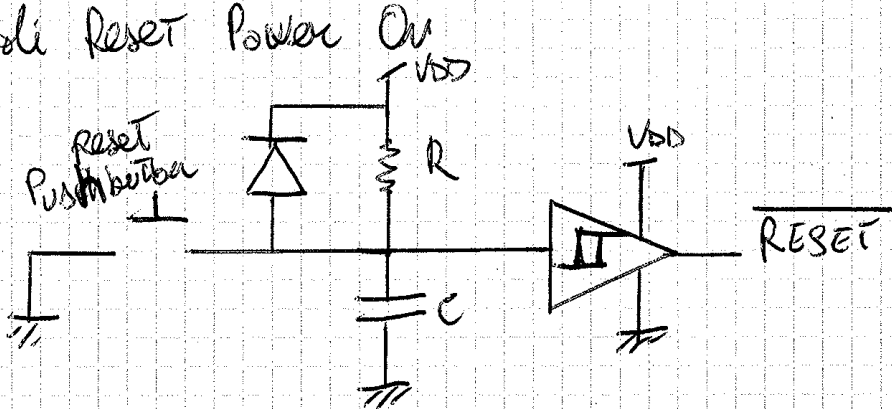
13/10/2013

Lo stato di Reset deve essere ben settato per essere sicuro da la massima parte bene.

- Reset legato al Power On, ovvero dipendentemente dalle ~~transistor~~ capacità parassite e non posso da 0 ad 1 bruscamente.

In generale si lavora con i CMOS, si hanno dei transistor parassiti, se durante l'accensione il segnale di ingresso supera la tensione di alimentazione i transistor parassiti si attivano e si ha il problema del latched-up.

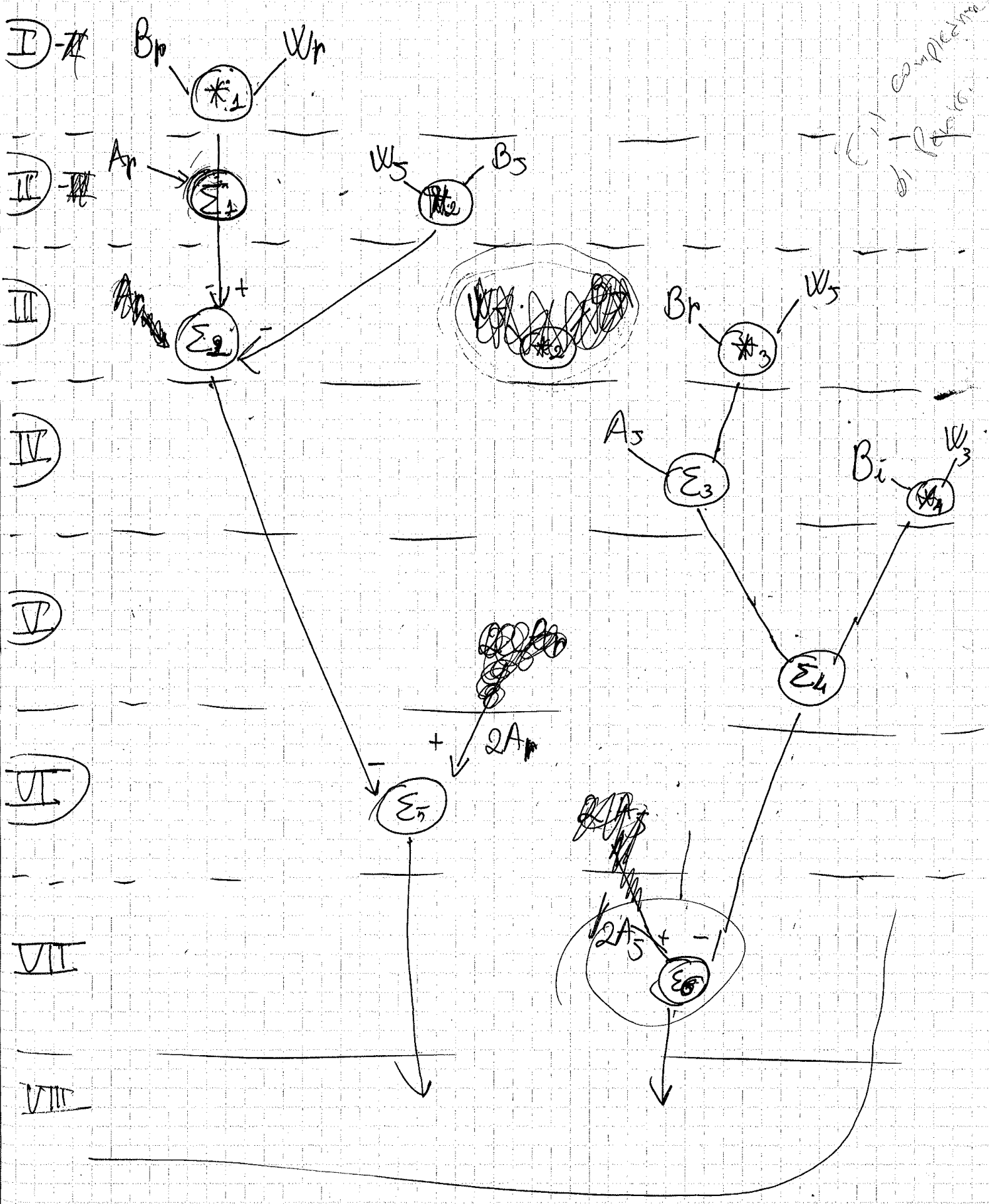
Circuito di Reset Power On



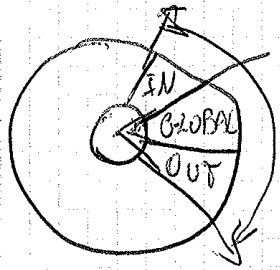
Una funzione del diodo è quella di evitare che il circuito veda il latched-up (poiché non si deve superare la V_{DD})

~~Si vuole questa avere un circuito~~

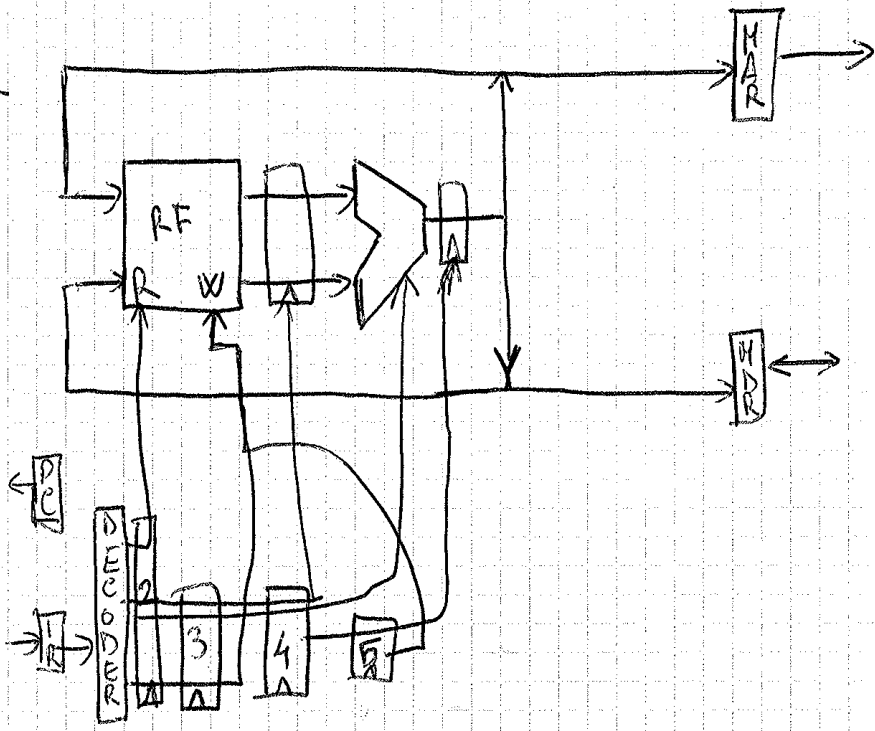
di registri



completed
 di per se

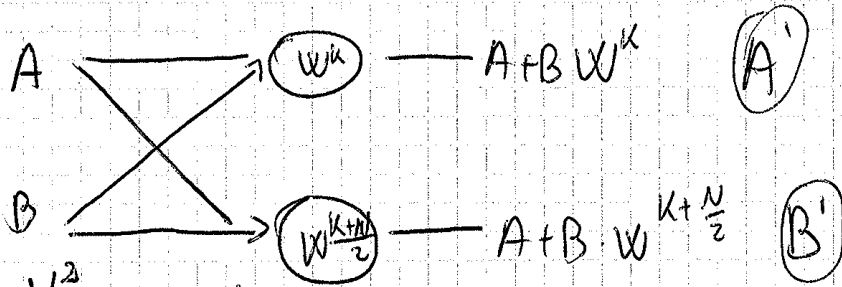


IN: registri di ingresso
 GLOBAL: disponibili nel progetto
 OUT: registri di uscita



FFT - Butterfly

$$\sum_{m=0}^{N-1} X_m e^{-j2\pi/Nmk}$$



Ordine di N^2 operazioni

Parte reale W_r

Parte immaginaria W_i

~~$$A' = A + W B = (A_r + j A_i) + W (B_r + j B_i) = A_r B_r - A_i B_i + j (A_i B_r + B_i A_r)$$

$$(A_r + j A_i) + (W_r + j W_i) (B_r + j B_i) =$$

$$= \underbrace{(A_r + W_r B_r - W_i B_i)}_{A'_r} + j \underbrace{(A_i + B_r W_i + B_i W_r)}_{A'_i}$$~~

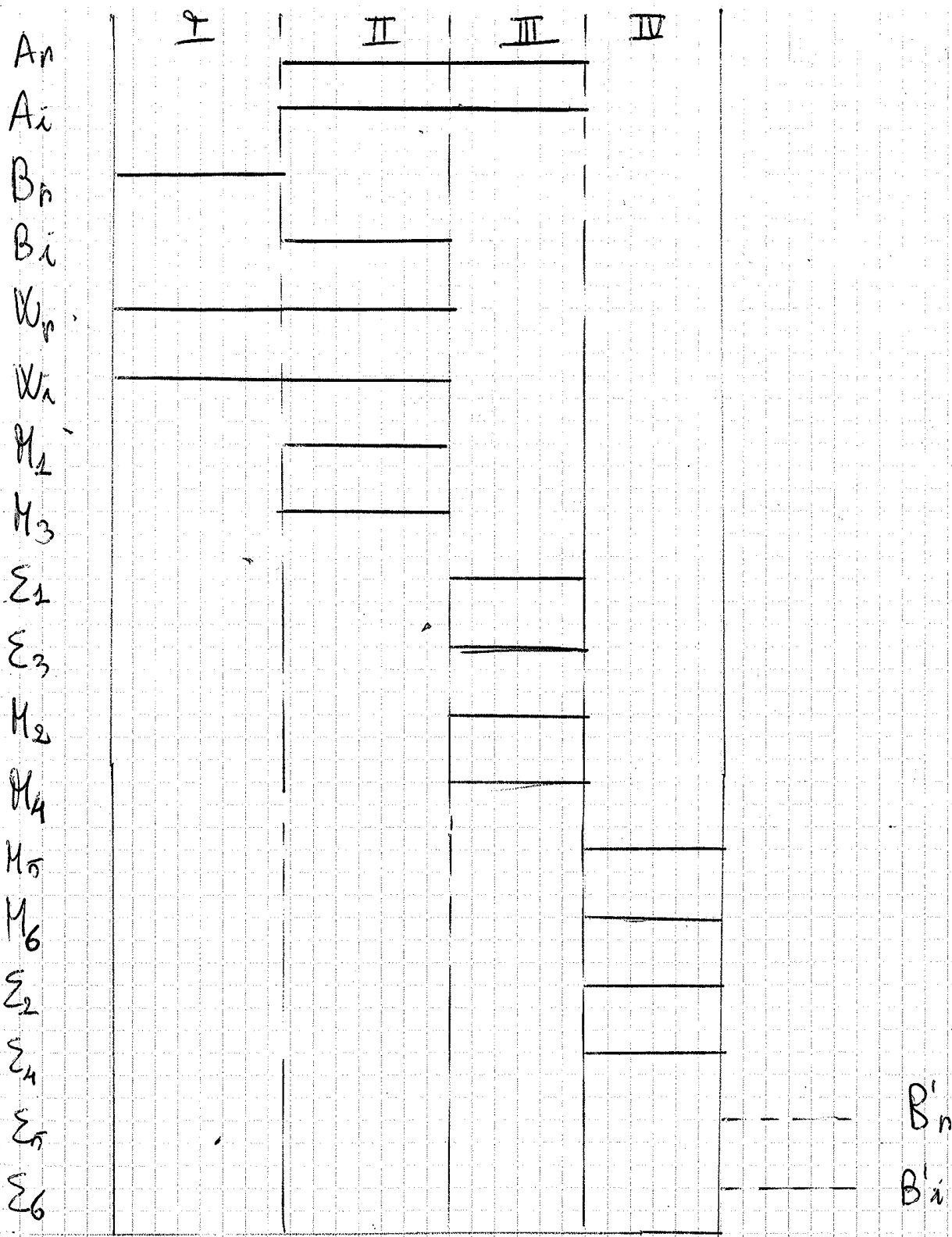
$$B' = A - W B = (A_r + j A_i) - (W_r + j W_i) (B_r + j B_i) =$$

$$(A_r - W_r B_r + W_i B_i) + j (A_i - B_r W_i - B_i W_r)$$

$$B' = \underbrace{2A_r - A'_r}_{B'_r} + j \underbrace{2A_i - A'_i}_{B'_i}$$

4 moltiplicazioni, 2 shift, 6 somme

- | | | |
|-----------------|-----------------------------|----------------------------|
| $B_r W_r = M_1$ | $\Sigma_1 = A_r + M_1$ | |
| $B_i W_i = M_2$ | $\Sigma_2 = \Sigma_1 - M_2$ | 1° dato in uscita da serie |
| $B_r W_i = M_3$ | $\Sigma_3 = A_i + M_3$ | |
| $B_i W_r = M_4$ | $\Sigma_4 = \Sigma_3 + M_4$ | 2° dato in uscita da serie |
| $2A_r = M_5$ | $\Sigma_5 = M_5 - \Sigma_2$ | 3° |
| $2A_i = M_6$ | $\Sigma_6 = M_6 - \Sigma_4$ | 4° |



Non sono necessarie 12 variabili perché molte variabili "vivono" per un colpo di clock e poi "muoiono" e possono essere condivise con una variabile fisica. Per le variabili temporanee ne le cavo con n variabili


Unconditional block floating point scheduling shift e aumento il fattore di scale.

Oppure uso i bit di guardia, si riduce il numero di ingressi. I bit di guardia devono essere usati per garantire che non vi sia overflow

Rounding: come hardware si usa un sommatore che sommi $0,5$. Max errore di arrotond. $\frac{1}{2} LSB$.

Round alternate: unta strada, alternativamente si va in quella più grande o quella più piccola. Il rounding viene in termini di ore e prestazioni.

Si usa allora l'overflow tramite ROM che fornisce le codici-table in cui entro con il linee di indirizzo e si vuole arrotondare l'ultimo bit) ed esce con $C-1$ linee di dati. Usando una ROM come tabella mi permette per ogni combinazione degli ingressi di definire cosa voglio avere in uscita. Per ogni valore degli ingressi si trova q , ovvero il valore che voglio. Il vantaggio è che il tempo è il tempo solo di lettura della ROM. A seconda di cosa scrivo nelle ROM uso le varie tecniche di rounding. In questa tecnica permette di avere l'arrotondamento da voglio trovare in un caso: quando entro con tutti 1 perché il risultato dovrebbe essere un numero su un bit che non è entrato dentro la ROM. ~~Un vantaggio~~ Se lo il arrotondamento ~~è~~ migliore è il numero di bit usati a meno quello errore.

	<u>I°</u>	<u>II°</u>	<u>III°</u>	<u>IV°</u>	<u>V°</u>	<u>VI</u>	<u>VII</u>	<u>VIII</u>
A_p					M			
A_i		Σ						
B_p	Σ		M	Σ		M		
B_i		M		M				
W_p	M			M				
W_i		M	M					
H_1								
Σ_1								
H_2								
Σ_2						A'_p		
H_3								
Σ_3								
H_4					A	A'_i		
Σ_4								
H_5								
Σ_5								B'_p
H_6								
Σ_6								B'_i

Numero massimo di registri impiegati contemporaneamente
4!

REG
VARIABLE

μ_2
 μ_1, μ_2, μ_3

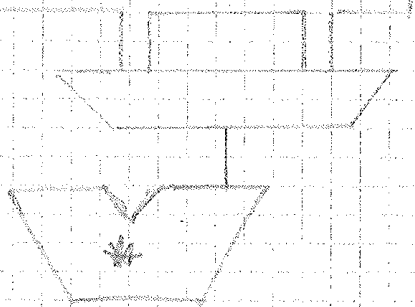
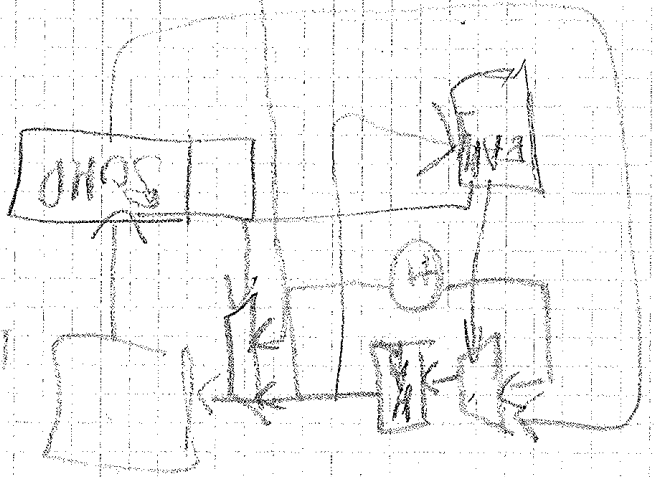
μ_4
 μ_4, μ_5, μ_6

Σ_2
 $\Sigma_1, \Sigma_2, \Sigma_3$

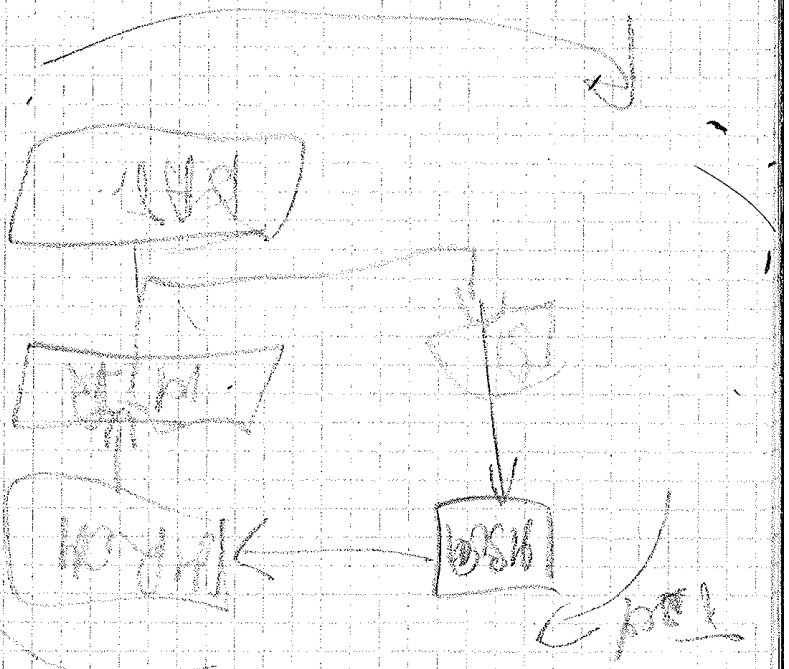
Σ_4
 $\Sigma_4, \Sigma_5, \Sigma_6$

REG
VAR.

REG
VAR.



STATUS
TOP



$\mu_1 = \mu_2$



$\mu_1 = \mu_2$

REF
FILE

6

