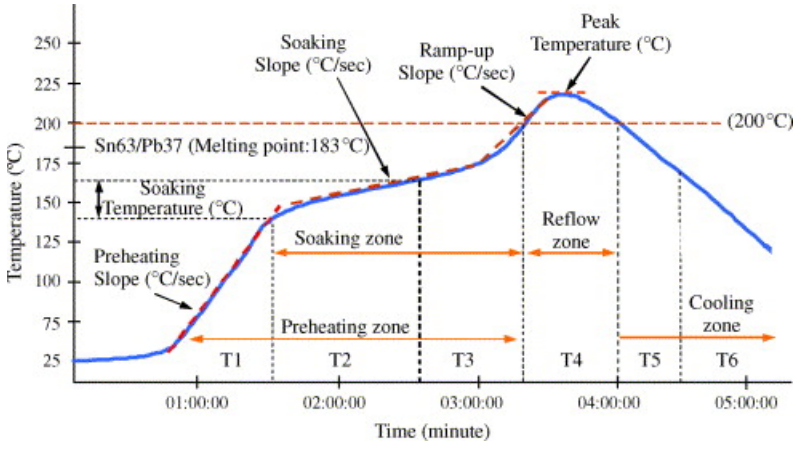
Kirkegaard Winkler (Kirkegaard)

MELTDOWN BRAIN II

23 June 2012

La cucina è di per sé scienza. Sta al cuoco farla divenire arte. (Gualtiero Marchesi)

Il deux ex machina per la pantomima dei processori è il cuoco. Lui passa tutta la vita ad eseguire pedissequamente il software suo o di altri, più o meno strutturato. Il suo vero scopo però non è solo quello di ottenere la finalità della ricetta. All'interno della cucina di un ristorante diventa un vero sistema operativo multitasking, con processi concorrenti, in real time e con il processo di ottimizzazione delle risorse sempre attivo. Le ricette devono essere eseguite correttamente, il rispetto della sequenzialità degli stati è parte stessa della ricetta e non può essere violata. Il cuoco ha strumenti (device) che elaborano gli elementi della ricetta trasformandoli in succulente uscite. Le materie prime binarie vengono elaborate in varie fasi, includendo in molti casi il processo di cottura. La curva di riscaldamento per la cottura delle pietanze nei forni della cucina, nel dettaglio seguente, è simile a quella impostata nei forni per la saldatura dei circuiti stampati. Al termine i nostri ingredienti sono pronti per il tocco finale del maestro, l'impiattamento.



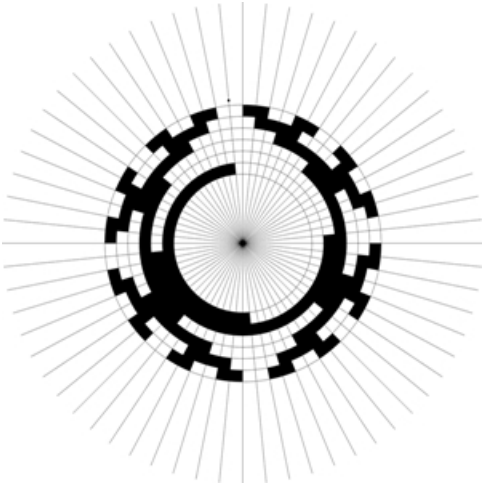
1-s2.0-S0957417404001721-gr2.jpg

Il grafico è un profilo termico d'esempio per le saldature al piombo. Attualmente le saldature senza piombo hanno un **melting point** più alto di 5-20 gradi a seconda della lega utilizzata.

E’ l’una di notte ho appena terminato il test della macchina a stati e quello che vedevo non era nulla di buono e cioè non vedevo nulla. Nulla di macroscopicamente evidente, nulla che mi desse un minimo appiglio, ma l'ora è tarda forse devo smetterla.
Ovviamente in questi sistemi embedded non abbiamo emulatori, così abbiamo implementato una prom con un boot che permette il download del binario sulla flash.Il progetto veniva compilato e l’uscita era un file oggetto intermedio in formato Motorola. I compilatori creano questi file intermedi di tipo ascii che venivano letti e convertiti dai programmatori di PROM. Esistono una marea di questi formati intermedi. I più conosciuti sono

- Motorola SRec
- Intel Hex

Con l’avvento degli ICE (in circuit emulator) il formato di uscita poteva diventare un file binario secco così i formati intermedi si sono un po’ persi...
Persi... come perso ero io in quel momento... riverso su quella scrivania con i faldoni delle specifiche di protocollo da una parte, gli schemi elettrici in formato A3 in ordine sparso, con i postit appicicati nello stesso ordine, le correzioni evidenziate in: verde mela, giallo acido e fuxia shoking, i grafici delle tensioni sul convertitore, il programma della GAL con le ottimizzazioni di Karnaugh. Uno vero tsunami di dati informazioni che avrebbe annegato chiunque.
A fianco un encoder in **codice Gray** ...o forse non era un encoder ma solo un selettore.. non lo ricordo più.



GrayEncoder.jpg

Sposto la luce della lampada. La scheda l’aveva rifatto non aveva inviato l’sms... allora mi fermo e riincomincio d’accapo. Guardo lo storico delle patch piuttosto lungo.
Bene, allora eravamo arrivati alla 2.34 prendiamo la 2.30 e carichiamola su.

Scarico il programma sulla flash. Resetto la scheda imposto la macchina e spedisco l'sms... non va.Proviamo la 2.20. Riscarico il programma sulla flash. Riresetto la scheda imposto la macchina e spedisco l'sms...
Funziona !
Bene !
Ho scoperto che il problema è nel firmware. Adesso devo solo capire quali sono state le modifiche fra la 2.20 e la 2.30. Semplice no ?
"
I linguaggi si suddividevano in 3 livelli ma di fatto la suddivisione ancora adesso non è cambiata.

- Low level
- Middle level
- High level

Si consideravano linguaggi di programmazione **low level** i linguaggi assembly di tutti i microprocessori.In generale il formato assembly era ed è ancora omogeneo. Il linguaggio è di fatto una sequenza di istruzioni macchina composti tipicamente da:

- Etichetta
- Operatore
- Operando

Nel linguaggio assembly non vi è nessun controllo di tipo perchè non esistono **tipi di dato**. Si considerano migliori quelli cosidetti **ortogonali** definendo con questo affascinante termine quelli che appunto hanno un set essenziale di istruzioni e molte modalità di indirizzamento.
Nell'esempio seguente un ordinamento "bubble sort" di una lista di numeri scritto in **assembly Z80**, si noti che l'assemblatore non ha bisogno di un terminatore dell'istruzione e che il ; è utilizzato per definire l'inzio del commento.

```
ORG 0000h
          LD IX,0D000h   ; PUNTATORE ALLA LISTA DEI NUMERI DA ORDINARE
          LD B,10        ; CONTATORE PRINCIPALE
          DEC B
          LD C,B         ; C è IL CONTATORE DEI CONFRONTI
ET2:      LD A,(IX+0)    ;CARICA IL PRIMO NUMERO DELLA COPPIA DA CONFRONTARE
          CP A,(IX+1)
          JP M,AVANTI
;SE IL PRIMO NUMERO NON è PIù GRANDE DEL SECONDO SI SALTANO
;LE ISTRUZIONI PER EFFETTUARE LO SCAMBIO
          LD D,A
          LD A,(IX+1)
          LD (IX+0),A
```

```
          LD (IX+1),D
AVANTI:   INC IX
          DEC C
          JP NZ,ET2
          DEC B
          JP Z, FINE
          LD C, B
          LD IX,0d000H
          JP ET2 FINE:
          HALT
```

L'indirizzamento è il modo con cui avviene l'accesso al dato. Nei microprocessori moderni le modalità di indirizzamento sono diverse...

- Implicito
- Diretto
- Immediato
- Assoluto
- Indiretto con registro
- Indiretto con spiazzamento
- Relativo
- Predecrementale
- Postdecrementale

...e così via in un florilegio di differenziate opzioni in grado di soddisfare il più istrionico NERD.
Nei linguaggi di medio e alto livello tutto questo si perde consentendo al programmatore di dimenticarsi delle difficoltà del sistema. Con il termine **Middle level** si consideravano i linguaggi come il **C**, l'esoterico **Forth** o ad esempio l'ancestrale e defunto **Occam**. Quest'ultimo poi era nato con l'ambizione e l'illusione di avere proprie le proprietà pragmatiche e di sintesi del filosofo (?!). Questi linguaggi erano in grado di interfacciarsi direttamente con il processore con in più la possibilità di creare tipi di dato. Questo permetteva di ottenere programmi decisamente più complessi (nel senso dalle finalità più complesse) con il nuovo approccio che era definito **programmazione procedurale**.
Esempio di codice C ancora l'ordinamento Bubble Sort di un vettore di numeri.

```
for (c = 0 ; c < ( n - 1 ); c++)
{
        for (d = 0 ; d < n - c - 1; d++)
       {
                    if (array[d] > array[d+1]) /* For decreasing order use < */
                    {
                              swap = array[d];
```

```
                                array[d]   = array[d+1];
                                array[d+1] = swap;
                    }
            }
}
```

Esempio bubble sort scritto in codice Forth :

```
bubble ( addr len -- )
 begin
 1- 2dup  true -rot  ( sorted addr len-1 )
 cells bounds ?do
 i 2@ bubble-test if
  i 2@ swap i 2!
  drop false   ( mark unsorted )
 then
  cell +loop  ( sorted )
 until 2drop ;
```

Esempio codice Occam.
Occam era un linguaggio scritto apposta per sistemi multiprocesso. Si noti che come il Basic non ha bisogno del terminatore di linea del codice. Si noti anche l'istruzione SEQ utilizzata per dire al compilatore che le istruzioni seguenti devono essere eseguite sequenzialmente.

```
--
-- Sorting Systolic Array        --
PAR J = 0
FOR MAXLENGTH
 BYTE x, y:
 SEQ
   sortchar[J] ? x
   WHILE x <> 255
     SEQ
       sortchar[J] ? y
       -- Send the smaller characters down the pipeline
       IF
         x < y
         SEQ
           sortchar[J+1] ! x
           x := y
        TRUE
        sortchar[J+1] ! y
-- After the WHILE loop has terminated, flush the pipeline.
```

```
   IF
    J < (MAXLENGTH - 1)
    sortchar[J+1] ! 255
    TRUE
SKIP
```

Gli **High Level** erano i linguaggi come il **Basic**, **Fortran**, **Cobol**, **RPG** o il **c++** e in ultimo momento poi si aggiunse un outsider il Java object oriented. In questi sistemi l'interfaccia con il processore si è definitivamente persa è il programmatore poteva finalmente concentrarsi solo sull'applicazione. I linguaggi High Level con l'avvento di **Java** hanno avuto negli ultimi anni un forte sviluppo. Uno dei difetti principali che venivano imputati al linguaggio Basic (ad esempio) era quello di essere interpretato. Le prestazioni di un programma Basic erano fortemente condizionate da questa operazione intermedia che veniva effettuata dal nostro sistema. I processori aumentarono le prestazioni e i sistemi operativi aumentarono esponenzialmente di complessità. Java sembra non essere in grado di reggere la concorrenza dei nuovi arrivati e super blasonati.

- Perl
- Pyton
- Php

Tutti questi linguaggi pur essendo infatti interpretati non soffrono di ansia da prestazione, anzi.
Esempio codice Perl.

```
#!/usr/local/bin/perl
#
# - Prompts for a file to read in.
# - Reads the file into an array.
# - Writes file to STDOUT with the lines numbered.
#
print "Please enter the file to read in (e.g., .login): ";
$filename = <STDIN>;
chop($filename);
# removing the newline character
open(IN, "$filename") || die "Could not open $filename...\n";
$i = 0;
while (<IN>)
{
    print ++$i, " : $_";
}
close(IN);
```

Esempio codice Pyton.
Dal linguaggio Occam descrtitto precedetente il **Pyton** prende una funzionalità interessante utilizzata nella implemntazione dei cicli. Come si può notare, non esistono terminatori di riga ne parentesi di definizione del blocco di istruzioni. Come il suo antesignano, nel Pyton, viene utilizzata l'**indentazione** per definire le istruzioni che devono essere eseguite nel ciclo. Questo obbliga il programmatore a scrivere automaticamente il codice, in modo molto leggibile e strutturato.

```
for letter in 'Python':
# First Example
if letter == 'h':
 break
print 'Current Letter :', letter
var = 10

# Second Example
while var > 0:
  print 'Current variable value :', var
  var = var -1
  if var == 5:
  break
print "Good bye!"
```

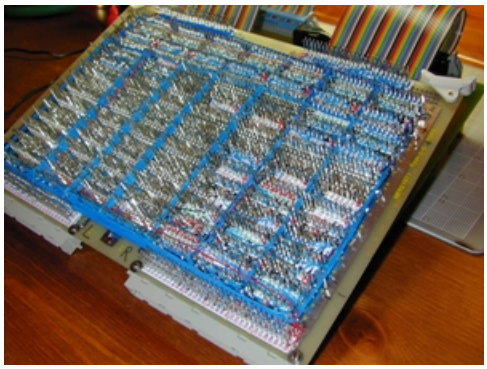
Esempio codice php.
Si noti il campo <?php ?> che permette al interprete detto anche "server sided" di riconoscere la sezione di codice php inserito all'interno della pagina html.

```
#!/usr/bin/php -q A Program to print a multiplication table
<?php print "Please enter a number ";
$fh = fopen("php://stdin","r");
$value = trim(fgets($fh,1024));
if ($value > 0)
{
    print "Here is a $value times table\n";
    for ($times=0; $times<13; $times++)
       {
           $result = $times * $value;
           print "$times times $value is $result\n";
        }
}
else
{
     print "You entered $value which is too small\n";
} ?>
```

Prendo la sonda dell’oscilloscopio e sposto il pulsantino della sensibilità sul 10x. Il cambio di range mi varia la capacità della sonda, rendendola meno invadente sul circuito. Tipicamente, hanno delle capacità da qualche decina di picofarad, sufficienti per innescare un qualsiasi quarzo da una decina di megaherz.
In rari casi, misurando appunto frequenze elevate come gli spike o i glich, la sonda potrebbe farli sparire, così per precauzione, evito a prescindere rari casi. Guardo a fianco la scheda di interfaccia della seriale e c’erano gli integrati 485 TH (Trough Hole) montati su una basetta mille fori.
Guardo l'alimentazione durante la registrazione del gsm e l'invio dell'sms sembra essere corretto. Nel momento che la radio trasmette, il livello di tensione traballa un pochino, per il carico aggiunto, ma non vedo niente di preoccupante.
Guardo poi tutta la sequenza di accensione e l'attivazione dell'**ignition**, anche li i tempi sono corretti e ampiamente sovrabbondanti per le specifiche del gsm utilizzato.
Un occhiata poi sommaria alla linea seriale che è pulita.
Ai nostri tempi la progettazione e la realizzazione di circuiti stampati era molto dispendiosa. Esisteva però, una teconologia alternativa che veniva utilizzata per la creazione di prototipi a basso costo. Questa tecnologia alternativa aveva caratteristiche che erano però considerate nettamente migliori della produzione dei circuiti stampati. Ad esempio, alcune normative militari pretendevano che alcune sezioni dei sistemi elettronici, venissero sviluppate in certo modo. Così nacque uno stuolo di personaggi che diventarono dei maestri in questa tecnica. Le schede fatte da loro erano delle vere opere d’arte di pregiatissima fattura.
Il nome ? Semplice, **Wire Wrap**!

WireWrapKit.jpg

</immagine>



fp_wire_wrap_side.jpg

Mi alzai stirandomi la schiena... *sono un po' indeciso se continuare oppure no... meglio di no sono arrivato alla fine delle mie possibilità fose è meglio che torno a casa*. Spesi le luci ed uscii, attraversando il lungo corridorio e poi scendendeno le scale della palazzina. Sono le due di notte e l'aria è piacevolemente fresca.

Ragiona Kirk. Ragiona. Usa la testa. Suddividi i problemi. Dividendo et imperando. Delegando il più possibile ed anche l'impossibile, nella speranza che qualcuno ci caschi.... Seleziona gli eventi con un dicotomico approccio senza ricorsioni. Sviluppa la tua logica con un numero sufficiente di variabili e col massimo di risorse disponibili. Perchè infatti devi complicarti la vita con affascinanti ottimizzazioni. Non serve, è pericoloso e alla fine rischi di essere tu il selezionato.

```
int ricercaBinaria(int lista[], int n, int x)
{
    int p,u,m;
    p = 0;
    u = n-1;
    while(p<=u)
    {
        m = (p+u)/2;
        if(lista[m]==x)
            return m;      // valore x trovato alla posizione m
        if(lista[m]<x)
            p = m+1;
        else
            u = m-1;
    }
  return -1;
}
```

Mentre sono in macchina penso a tutte le modifiche fatte nell'ultimo periodo. Il codice aggiunto ma anche al codice tolto. Poi arrivo a casa mi faccio un cocktail di birra e whisky, mi butto nel divano e durante il volo entro in uno stato di incoscenza.

8:45 del giorno dopo

Squillò il telefono... guardai l'orologio senza riuscire a capire che ora era ma intuivo che fosse tardissimo. Girandomi mi appoggiai sul tavolino e la bottiglia di birra assieme al bicchiere si rovesciarono cadendo, ovviamente salvai solo la bottiglia di birra vuota, mentre il bicchiere si frantumò nel pavimento. Mia madre si sporse guardandomi sconsolata e mi vide inciampare sulla colonna di Zagor che avevo a fianco del tavolino da fumo.

Grugnisco un frase del tipo "mamma per piacere mi dici che ora è ?"

e dalla bocca uscii un "mm ppr dchh straminc chhrr"

"Scusa che hai detto ?"

Intanto il telefono continuava a squillare e il cervello mi entrò in risonanza con la suoneria.Guardai il cellulare è vidi sul display il nome del chiamante ed era ancora lui Zio Flubbert !!!!!!!

"Pronto zio... nooo non posso venire sono nei casini... questa sera vengo. Non ti preoccupare che questa sera vengo... si ti dico che vengo..."

Tentai di uscire senza fare colazione ma non ci riuscii. Se facevo una cosa del genere rischiavo un incidente diplomatico famigliare peggiore del disastro sul lavoro. E così ingurgitai una tazza di caffellatte con sedici marie che avevano una temperatura prossima alla liquefazione della materia. Uscii bofonchiando con la mano sulla bocca ustionata e con mia madre che mi diceva

"Amoreee.. Non fare tardi stasera che devi passare da zio Flubbert"