

# ISTITUTO NAZIONALE DI FISICA NUCLEARE

Centro Nazionale Analisi Fotogrammi

---

**INFN/TC-87/8**

15 Giugno 1987

A. Castelvetti e E. Remiddi:

**REALIZZAZIONE DI SCHEDE ELETTRONICHE CON TECNICHE DI  
WIREWRAP E MULTIWIRING**

**REALIZZAZIONE DI SCHEDE ELETTRONICHE CON TECNICHE DI  
WIREWRAP E MULTIWIRING**

*Autori: A. Castelvetti, E. Remiddi*

Introduzione

Con la diffusione del CAD (Computer Aided Design) per la progettazione di schede elettroniche aumenta l'importanza delle tecniche di tipo digitale che ne permettano la rapida realizzazione in maniera virtualmente priva di errori: tra queste, principalmente, il Wire Wrap per prototipi o piccolissime serie e il Multiwire (nome depositato della Kollmorgen Company) per piccole o medie produzioni.

Da questo punto di vista il Multilayer, che probabilmente resta la migliore tecnica per produzioni di media o grande serie, richiede una elaborazione più complessa, non sempre meramente digitale e che può quindi comportare tempi di realizzazione maggiori nella messa a punto o per piccole modifiche di progetto.

Nell'ambito della collaborazione APE (Array Processor Emulator) il CNAF, dopo un breve ricorso alle risorse per il Wire Wrap disponibili al CERN, si è dotato di una macchina semiautomatica di Wire Wrap; successivamente, sempre seguendo ed in parte utilizzando procedure in uso al CERN, ha messo a punto e completato nuove originali procedure per la realizzazione di schede WW o MW, che partendo dal solo progetto CAD generano, nei formati richiesti dalle ditte operanti nel settore, tutta l'informazione necessaria ai vari stadi di produzione di una scheda (lucidi per circuiti stampati, foratura, rifiniture varie e, naturalmente, filatura in WW o MW).

La procedura è già stata utilizzata per la realizzazione della prima scheda in MW prodotta in Italia: la scheda di memoria dinamica da 16 megabytes per APE.

L'insieme delle procedure e del software sviluppato è accessibile nell'ambito INFN via INFNET a tutti gli interessati; il CNAF è disponibile a offrire tutta l'assistenza del caso per la sua utilizzazione, come pure a correggere eventuali errori od apportare miglioramenti al software sviluppato su richiesta degli utilizzatori.

La macchina WW del CNAF è pure a disposizione degli interessati; la macchina MW cui si fa riferimento è quella della Ditta CORONA di Leinì (Torino) operativa dal dicembre 1986.

## DESCRIZIONE DI MASSIMA DEL TIPO DI ELABORAZIONE DA COMPIERE E DEI PROGRAMMI DA USARE NELLE VARIE FASI

### DAL CAD AL WW .....

Si supponga di aver completato il progetto CAD; i files con la lista dei collegamenti etc. vanno trasferiti su VAX.

Una prima serie di programmi riscrive l'output del CAD (che nel caso prima citato era il CASE CT1000 su PC IBM) nel formato utilizzato nel seguito e ispirato al formato richiesto dal programma WRAP4 del CERN). Per procedere, occorre fornire, in aggiunta, la documentazione per il layout (posizione e descrizione dei componenti), che va preparata a mano indipendentemente dal CAD.

Il programma KLUDGE usa la documentazione di layout e produce principalmente una visualizzazione del layout della scheda da realizzare (kludge board) assieme a liste delle coordinate di pins, fori, etc. da utilizzare poi per la generazione dei lucidi, foratura della scheda WW e forature non standard per schede MW.

I programmi NETTR (NET-TRanslation) e NETMERGE riformattano i files dei collegamenti usciti dal CAD al formato richiesto dal successivo programma WRAP. (L'uso di un CAD diverso richiederebbe l'adattamento di NETTR e NETMERGE)

Il programma WRAP usa gli schemi dei collegamenti logici e la documentazione del layout per generare collegamenti tra pins fisici sulla scheda, sia nel formato richiesto dal programma che comanda la macchina WW, sia in quello richiesto dalla successiva elaborazione per il MW.

La filatura vera e propria sulla macchina OK-machine WWM-600 è controllata da un ulteriore programma che gira sul VAX e traduce l'output di WRAP in comandi per la macchina, interfacciata al VAX da un terminale semi-intelligente.

### ..... AL MW

L'input per l'elaborazione è dato dai files per il MW prodotti da WRAP; l'elaborazione vera e propria avviene con software originale PCK, accessibile via rete INFNET-HEPNET sul VAX VXCRNA del CERN.

L'output finale è costituito da un certo numero di files su floppy-disc da 8 pollici formato DEC-RT11, contenenti i comandi per il PDP-11 che controlla la macchina del MW.

È ovviamente consigliato visualizzare il contenuto dei floppies su terminale grafico o plotter prima di passare alla produzione.

## PRODUZIONE DI LUCIDI PER CIRCUITI STAMPATI

La produzione di lucidi per realizzare i piani interni di terra e alimentazione, per stampe serigrafiche, etc, se necessaria, si realizza fornendo al costruttore alcuni files sul supporto fisico richiesto dal photoplotter. I files vengono generati da alcuni programmi specifici che utilizzano come input i dati forniti dal programma KLUDGE.

## DESCRIZIONE DEI SINGOLI PASSI DELLA PROCEDURA (input, funzionamento, output)

### VISIONE D'INSIEME:

In fig. 1 è riportato il flow-diagram completo della procedura.

I passi A1) - A5) riguardano il passaggio dal CAD alla produzione WW o dell'input per MW.

I passi B1) - B5) riguardano l'esecuzione della procedura MDS2, che genera gli ordini per la filatura in MW e la perforazione delle schede.

C1) - C3) riguardano le piccole elaborazioni aggiuntive, non effettuate dal software precedente, richieste per la realizzazione effettiva delle schede, la visualizzazione dei dati, e il trasferimento delle informazioni al costruttore.

Tutti i files sono di tipo ASCII, di lettura immediata per verifiche e controlli.

## A1-A5 - GENERAZIONE DEI DATI PER LA PROCEDURA MDS2 E WIRE-WRAP

### A1) CAD

Non ha importanza il CAD utilizzato per il progetto elettronico. Si richiede solo che i files con le liste dei collegamenti logici (detti di tipo .TBL) possano essere trasferiti (via floppies, via telematica, ecc.) sul VAX su cui risiedono i programmi successivi.

Sarà compito dei programmi di reformatting, facilmente modificabili, adattare l'output CAD in un input compatibile al software successivo.

Per sistemi CAD installati su piccoli PC può essere conveniente spezzare il progetto in più parti: si guadagna tempo nel caso di modifiche che riguardano solo una parte del progetto.

Non tutti i files prodotti dal CAD sono necessari per l'elaborazione successiva : per il CAD CT1000 CASE su PC IBM, per esempio, solo i files .TBL vanno trasferiti su VAX.

### A2) - Reformatting

Tre piccoli programmi usati in cascata si incaricano di tradurre le informazioni contenute nei singoli files .TBL in un input adatto al successivo programma WRAP. In particolare:

il programma NETTR genera, per ogni file .TBL :

- 1) il netlist file (.NET) che è una sequenza di connessioni tra i pins nel formato usato nel seguito;
- 2) il Component types file, (.ICT) con le informazioni sul tipo di componenti usato.

il programma NETMERGE ricombina i files parziali .NET e .ICT generati da NETTR.

il programma CONDENSE rende più compatto il file .NET risultante da NETMERGE.

nel seguito si utilizzano solo i files .NET e .ICT uscenti da CONDENSE.

### A3) - ALTRI FILES PER WRAP

Al programma WRAP occorrono oltre ai files .NET e .ICT i files che contengono:

- 1) La posizione dei componenti sulla scheda (component-file .DAT);
- 2) La definizione (opzionale) dei pins terminatori per i segnali che richiedono la terminazione (file .TER);
- 3) La descrizione della piedinatura dei componenti usati (features file)
- 4) La definizione dei pins di terra e alimentazione per ogni tipo di ponente usato (file .GEV).

I files 1) e 2) sono specifici del progetto in esame e vanno preparati a mano; i files 3) e 4) vanno pure preparati a mano ma costituiscono una specie di libreria che può essere utilizzata per ogni progetto.

N.B.!!! I files 2), 3), 4) non vanno modificati se si modifica il file 1), e vanno aggiornati solo se si introducono nel progetto tipi di componenti precedentemente non previsti.

Concludendo: Il progettista dovrà occuparsi solo della creazione via editor del component-file, del file terminatori e, via CAD, del .TBL file.

#### A4) - WRAP: MW-WW-program

Il programma WRAP, (VXCNAF::DUA1:[CVETRI.WRAP]WRAP.EXE) è operativo da più di un anno ed è a disposizione di eventuali interessati.

Il programma WRAP è utilizzato per :

- 1) generare files di ordini per la wrappatrice OK WWM-600 ottimizzando il percorso dei segnali;
- 2) generare l'input per la procedura MW MDS2 installata al CERN.

Per l'esecuzione del programma si consiglia ad ogni utente di definire nel proprio LOGIN.COM il simbolo

```
WRAP:==run VXCNAF::DUA1:[CVETRI.WRAP]WRAP.EXE
```

o di copiare periodicamente tale .EXE file per avere la certezza di utilizzare sempre la versione più aggiornata.

Il programma, di uso interattivo e abbastanza veloce chiede :

- 1) il passo in mills (solitamente 100 mills)
- 2) la radice dei files (viene utilizzata per accedere ai files radice.NET, radice.DAT, radice.TER, radice.ICT, radice.GEV e per creare i files di output radice.LIS, radice.OFF, radice.CHK, radice.ERR, radice.HLS, radice.MWN);
- 3) il nome completo del features-file;
- 4) se si vuole utilizzare il file dei terminatori;
- 5) se si vuole o no l'output per il multiwiring:

#### IN CASO AFFERMATIVO

genera i files radice.HLS (holes) e radice.MWN (netfile formato MW)

#### IN CASO NEGATIVO

genera i files radice.LV1 e radice.LV2 che contengono gli ordini per la wrappatura su OK rispettivamente a livello 1 e 2.

#### LIMITI :

Il programma WRAP prevede di lavorare al massimo con:

- 200 records nel file .TER
- 150 records nel file .GEV
- 150 records nel file .ICT
- 1000 componenti nel file .DAT
- 50 features nel package file
- 150 piedini per feature
- 500 piedini per segnale
- 6500 fili da wrappare

L'aumento di queste dimensioni è banale (ma richiede la ricompilazione del programma WRAP).

WRAP effettua parecchi controlli incrociati sulla congruenza dei dati di progetto: in presenza di un errore "fatale" il programma si arresta e l'errore viene indicato. Altri avvertimenti vengono registrati nel file .ERR di cui si consiglia sempre la lettura.

In caso di errori di progettazione riscontrati dal programma WRAP, è consigliabile ripercorrere gli steps A1) e A2) o A3) dopo aver corretto i files sorgente, piuttosto che intervenire via editor sui files prodotti. Questo permette di avere sempre tutti i files corretti ad ogni livello.

#### A5) - WW-files

Note sul Wire-Wrap installato al CNAF:

La macchina OK WWM-600 del CNAF, che prevedeva di essere controllata via paper-tape, è stata modificata, ed ora è in sostanza una estensione di un terminale intelligente collegato al VAX 750 dove risiede il programma di gestione per la macchina stessa. È il programma di gestione su VAX che interpreta gli ordini contenuti sui files generati da WRAP, li trasforma in ordini formato "OK" e via terminale li invia alla WWM-600.

I files utilizzati dal programma di gestione sono il .LV1 e .LV2

#### B1-B5 - UTILIZZO DELLA PROCEDURA MDS2 INSTALLATA AL CERN

Per una descrizione abbastanza dettagliata e per l'uso di MDS2, che è una procedura installata al CERN sul VAX 8600, nodo VXCRNA della rete DECNET di cui fa parte INFNET, consultare il rapporto :

M.F. Letheren, CERN - DD : " Cern VAX/VMS command procedures for multiwire board layout ",

ottenibile dall'autore al CERN (reperibile anche come VXCRNA::MDT). Secondo tale rapporto, tutti gli input-files necessari devono avere la stessa radice, che viene chiamata 'JOB\_NAME'. Il 'JOB\_NAME' è la prima cosa che chiede MDS2, ad esso si riferisce per creare una sub-directory e per cercare gli input-files che sono:

JOB\_NAME.NET  
JOB\_NAME.HLS  
JOB\_NAME.OBD  
JOB\_NAME.OBM  
JOB\_NAME.SPC  
JOB\_NAME.SPL  
JOB\_NAME.BOR

I files di output saranno nella sub-directory 'job\_name' e si chiameranno tutti PCKxxx.DAT, dove xxx è un numero di tre cifre. MDS2 produce parecchie decine di tali files, ma alla fine ne servono solo dieci o dodici (vedi B5).

Rivolgersi alla documentazione citata per maggiori dettagli. L'esperienza farà il resto.

#### B1) - INPUT FILES PER MDS2

Il file Job\_name.HLS è generato da WRAP.

Il file Job\_name.NET è il file .MWN generato da WRAP, da ribattezzare .NET al momento del trasferimento su VXCRNA.

#### B2) - ALTRI INPUT FILES PER MDS2

I files .OBD, .OBM, .SPC, .SPL, .BOR vanno confezionati via editor secondo le specifiche riportate nella citata documentazione. Si tratta comunque di poche righe per file.

### B3) - CERN MDS2 MW procedure

L'utente che desidera utilizzare MDS2 dovrà innanzi tutto essere abilitato all'uso di VXCRNA e chiedere a M.F. Letheren la documentazione.

Dovrà inserire all'inizio del LOGIN.COM le seguenti definizioni :

```
$ ON ERROR THEN CONTINUE
$ ASS DISK$DD MDT$ROOT
$ @MDT$ROOT:[MDT.RUN.COM]MDTLOGIN
$ @MDT$ROOT:[MDT.CERN.COM]CERNDEF
$ IF "F$MODE()" .EQS. "BATCH" THEN EXIT
```

Dovrà inoltre disporre di una directory su VXDEV nella quale MDS2 andrà a scrivere i "manufacturing files" per poi registrarli su floppies (vedi paragrafo successivo) e di una directory su IBM per l'eventuale uso del plotter Versatec del CERN.

### B4) - TRASFERIMENTO FILES

La procedura MDS2 termina con il trasferimento dei manufacturing files su VXDEV nella subdirectory [user.JOB\_NAME] che viene automaticamente creata. Tutti i files sono costituiti da records di cinque halfwords e sono binari.

Entrando su VXDEV ed utilizzando il comando

```
@[MWIRE]FLOPPIES
```

si ottiene la registrazione dei files sui floppies di VXDEV via XFL. La stessa operazione può essere compiuta su un altro Vax utilizzando la utility standard VAX/VMS EXCHANGE :

```
$ EXCHANGE (ret)
EXCHANGE > INIT CSA1:/VOLUME_FORM=RT11
EXCHANGE > COPY filename CSA1:/RECORD_FORM=(FIXED=10)/VOL_FORM=RT11
EXCHANGE > ....idem per ogni filename
EXCHANGE > DIR CSA1: (per controllo)
EXCHANGE > EXIT
```

N.B.!!! - Attualmente il floppy disk formato RT11 è l'unico mezzo di interfacciamento con la ditta costruttrice.

### B5) - MANUFACTURING FILES PCKxxx.DAT

I files di cui al punto B4) sono i seguenti:

```
PCK001.DAT hole data from component side
PCK025.DAT hole data from soldering side
PCK016.DAT energiser data from component side
PCK018.DAT checker data from component side
PCK198.DAT test cluster holes from comp. side
PCK199.DAT test cluster wires from comp. side
PCK101.DAT level 1 wiring data (soldering side)
```

PCK111.DAT level 1 sifter data (soldering side)  
PCK201.DAT level 2 wiring data (component side)  
PCK211.DAT level 2 sifter data (component side)

Se esistono altri livelli di wiring saranno presenti anche i corrispondenti files PCKn01.dat e PCKn11.dat dove "n" è il livello di wiring.

## BRICOLAGE

### C1) - FILES ACCESSORI

I files PCKxxx.DAT forniti da MDS2 non sono sufficienti per la costruzione completa della scheda.

Occorre fornire alla ditta costruttrice le informazioni necessarie per la realizzazione dei piani interni di terra e alimentazione, per la perforazione e collegamenti tra pins a passo non standard (non previsti da MDS2), per la realizzazione di piste stampate in aggiunta ai collegamenti MW (se ve ne sono), per stampe serigrafiche sulla scheda e quant'altro si pensi di aggiungere.

Tutti questi dati vanno forniti su files identici come formato ai files binari PCKxxx.DAT, nello stesso sistema di riferimento.

Si tratta di files binari, cosiddetti "GD Files", costituiti da linee (records) di 5 words da 16 bits.

Per ottenerli occorre:

- 1) che i dati da registrare siano definiti su VAX come Halfword (16 bits)
- 2) che nella "OPEN" del file venga dichiarato :  
RECORDTYPE='FIXED',RECL=10,FORM='FORMATTED',CARRIAGECONTROL='NONE'
- 3) che l'istruzione di scrittura sia del tipo :  
WRITE (iunit,'(5A2)') list

Inoltre, per avere (senza entrare in dettaglio) una identità formale con i files PCKxxx.DAT occorre scrivere prima dei dati:

- 4) due records "dummy" di cinque Halfwords
- 5) un terzo record:  
maxrec,ixmin,iymin,ixmax,iymax  
dove maxrec > = al numero massimo di records del file e gli altri dati rappresentano valori < = ai minimi e > = ai massimi delle coordinate presenti
- 6) altri sei records "dummy"
- 7) con il decimo record devono iniziare le informazioni utili

Si riporta la descrizione del significato delle singole voci presenti nel file. Per "words" deve intendersi la parola da 16 bits.

### GD File Format

GD Files are graphical data files. Each record (line) has 5 words

A X Y B C

A is the type or process being done to the record

- 0 - inflection point
- 1 - pen down (for drawing lines)
- 2 - pen up (for drawing lines)



- 3 - hole or net point
- 4 - wired through hole
- 5 - wire start
- 6 - wire end
- 1 - end of file marker

X Y absolute X Y coordinates  
B hole size, net #, or line width  
C aperture size for photoplotter, or sequence #

GD file records:

1 - 9 Overhead information about file, including name  
End of file marker is           -1    32767    32767  
                                  A       X       Y

È necessario quindi realizzare espressamente dei programmetti più o meno complessi per produrre questi files, che dovranno poi essere registrati via EXCHANGE su floppies ed inviati alla ditta costruttrice.

Sarebbe inutilmente dispersivo descrivere i programmi all'uopo realizzati al CNAF. Si fa presente la necessità della congruenza nel formato dei files e nel sistema di riferimento.

## C2) - PLOTS DI CONTROLLO

Per controllare la congruenza tra i files PCK e quelli prodotti dai programmi di BRICOLAGE il plotter è indispensabile. Al CNAF viene usato un programma che permette di sovrapporre su di un solo disegno le informazioni provenienti da un numero qualsivoglia di files (analogo al plotter program per Versatec utilizzato da MDS2). Occorre solo fare attenzione a non mescolare files relativi al "component side" a files relativi al "soldering side".

## C3) - TRASFERIMENTO INFORMAZIONI ALLA DITTA COSTRUTTRICE

Attualmente i files di comandi pervengono alla ditta costruttrice su floppies. Sono in corso contatti per un auspicato collegamento telematico. I files per la filatura e la perforazione prodotti dall'utente vengono utilizzati dal costruttore direttamente senza alcuna rielaborazione. I files per l'uso del photoplotter vengono invece rielaborati dal costruttore: è previsto lo studio dei formati necessari al photoplotter per evitare la rielaborazione dei files e ridurre ulteriormente i tempi di realizzazione. (2 settimane tra l'arrivo dei floppies e la consegna delle schede).

### PRINCIPALI FILES RICHIESTI O GENERATI NELLE VARIE FASI, CON DESCRIZIONE DEL FORMATO ED ESEMPI

File .TBL (generato da CAD, input a NETTR)

formato:

segnale

componente    tipo    pin

N.B. : Il nome del segnale deve iniziare a colonna 1.

I nomi dei componenti devono essere preceduti esclusivamente da uno o più "blanks"

Esempio:

BEN0

U101	74LS374	1
U106	74LS374	1
U112	74LS374	1
U117	74LS374	1

CK120SL

U601	LS646	23
U602	LS646	23
U603	LS646	23
U604	LS646	23

.  
.  
.

etc., etc. fino all'EOF

File .NET (output di CONDENSE)

formato:

segnale	comp	pin	comp	pin	comp	pin	comp	pin
---------	------	-----	------	-----	------	-----	------	-----

Esempio:

A10	U217	9	U218	9	U631	2	U635	2
A10	U481	55	U491	55	U501	55	U511	55
A10	U521	55	U531	55	U541	55	U551	55
A10	U401	55	U411	55	U421	55	U431	55
A10	U441	55	U451	55	U461	55	U471	55
AEN0	U102	1	U105	1	U113	1	U116	1
AEN0	U129	1	U132	1	U135	1	U138	1
AEN0	U239	1	U220	12				
AP01	U239	19	U007	16				
AP02	U239	16	U007	14				
BEN1	U128	1	U133	1	U134	1	U139	1
BEN1	U220	18						

.  
.  
.  
.

etc., etc. fino all'EOF

File .ICT (output di CONDENSE)

formato:

tipo	comp	comp	comp	comp	comp	comp
------	------	------	------	------	------	------

esempio:

74LS374	U101	U106	U112	U117	U128	U133
74LS374	U134	U139	U102	U105	U113	U116
74LS374	U129	U132	U135	U138	U239	U118
74LS374	U204	U205				
LS646	U391	U392	U393	U394	U395	U396
LS646	U397	U398	U601	U602	U603	U604
LS646	U361	U365	U371	U374	U375	U378
LS646	U381	U384	U385	U388		
ALS632	U382	U386				
74LS280	U103	U104	U114	U115	U130	U131
74LS280	U136	U137				
74LS245	U346	U345	U344	U343		
74LS244	U217	U218	U216			

etc., etc. fino all'EOF

File .DAT (component-file, input a KLUUDGE, WRAP)

formato:

comp: feature x\_pin1 y\_pin1 magnif\_x magnif\_y rotazione

N.B. : I dati devono essere separati da uno o più blanks. Il carattere " : " che segue immediatamente il component-name e'obbligatorio, in analogia al formato previsto da WRAP4 del CERN.

esempio:

```
U001: ROW32 3300 100
U002: ROW32 3300 200
U003: ROW32 7200 100
U004: ROW32 7200 200
U005: ROW32 11900 100
U006: ROW32 11900 200
U007: ROW32 15800 100
U008: ROW32 15800 200
U009: ROW32 6000 11600 1 1 180
U010: ROW32G 6000 11800 1 1 180
U701: CON2 100 1400
U702: CON2 500 1400
```

.  
.
.

etc., etc. fino all'EOF

Feature-file (descrizione piedinature, input a KLUDGE, WRAP)

formato:

```
FEATURE feature_name
pin_ident: hole_code x y drill_code photopl._code !Comments
.
.
.
END feature_name
```

N.B. : I simboli FEATURE e END rispettivamente all'inizio e alla fine di ogni feature sono obbligatori, come pure il simbolo " : " dopo il pin\_ident. Gli eventuali commenti ! <G> , ! <V> , se esistono, identificano pins di terra e alimentazione.

esempio:

```
FEATURE ZEN2
1: 3 0 0 1 8
2: 3 0 -100 1 8
END ZEN2

FEATURE CON2
1: 3 0 0 1 8 ! <G>
2: 3 300 0 1 8 ! <V>
END CON2

FEATURE DIP20
1: 3 0 0 1 8
2: 3 0 100 1 8
3: 3 0 200 1 8
4: 3 0 300 1 8
5: 3 0 400 1 8
6: 3 0 500 1 8
7: 3 0 600 1 8
```

```
8:   3   0   700   1   8
9:   3   0   800   1   8
10:  3   0   900   1   8 | <G>
11:  3  -300  900   1   8
12:  3  -300  800   1   8
13:  3  -300  700   1   8
14:  3  -300  600   1   8
15:  3  -300  500   1   8
16:  3  -300  400   1   8
17:  3  -300  300   1   8
18:  3  -300  200   1   8
19:  3  -300  100   1   8
20:  3  -300   0    1   8 | <V>
END DIP20
```

etc., etc. fino all'EO

File .PIN (pin-file, output di KLUDGE)

formato : A, B, C, D, E

A : \*\*\* dummy\_line di commento, per facilitare la lettura

VCC pin di alimentazione

GND pin di terra

PIN pin diverso da terra e alimentazione

B :  
coordinata X

C :  
coordinata Y

D :  
codice per photoplotter

E :  
codice di perforazione

Esempio di file .PIN

```
***, riga # 160
***, riga # 159
VCC,15900,10900, 13, 1
VCC,15900, 8700, 13, 1
VCC,15900, 6500, 13, 1
VCC,15900, 4300, 13, 1
```

```
GND,15900, 2400, 13, 1
PIN,15900, 2300, 12, 1
PIN,15900, 2200, 12, 1
PIN,15900, 2100, 12, 1
PIN,15900, 2000, 12, 1
PIN,15900, 1900, 12, 1
PIN,15900, 1800, 12, 1
VCC,15900, 1700, 13, 1
GND,15900, 1300, 13, 1
PIN,15900, 1200, 12, 1
PIN,15900, 1100, 12, 1
PIN,15900, 1000, 12, 1
PIN,15900, 900, 12, 1
PIN,15900, 800, 12, 1
PIN,15900, 700, 12, 1
VCC,15900, 600, 13, 1
***, riga # 158
GND,15800,10900, 13, 1
GND,15800, 8700, 13, 1
GND,15800, 6500, 13, 1
GND,15800, 4300, 13, 1
```

.  
.  
.

etc., etc. fino all'EOF

File .TER (terminators-file, input a WRAP)

formato :    A    B    C    D    E

A : nome del segnale da terminare

B, C : componente, pin della prima terminazione

D, E : componente, pin della seconda terminazione

Esempio di file .TER :

```
A10    U635   2    U631   2
A11    U635   3    U631   3
A12    U635   4    U631   4
A13    U635   5    U631   5
A14    U635   6    U631   6
A15    U635   7    U631   7
A16    U636   2    U632   2
A17    U636   3    U632   3
A18    U636   4    U632   4
A19    U636   5    U632   5
```

.  
.

etc., etc. fino all'EOF

File .GEV (Ground-Vcc-File, input a WRAP)

Formato :    A    B    C

A : tipo di componente

B : pin di alimentazione

C : pin di terra

N.B:

se B = XXX alimentazione non prevista

se C = XXX terra non prevista

Esempio di file .GEV

74LS379	16	8
74LS378	16	8
74LS534	20	10
74LS175	16	8
74LS373	20	10
74LS190	16	8
74LS240	20	10
74LS245	20	10
74LS181	24	12
74LS182	16	8
74LS381	20	10
74LS382	20	10
74LS283	16	8
74LS280	14	7
74F350	16	8
CON	2	1
CAP	XXX	1
RES	XXX	1
CON32	XXX	10
TR220	XXX	1
LED	4	XXX
DIODE	2	XXX

.  
.  
etc.,etc. fino all'EOF

Files .LV1, [.LV2] (Wrap-files level 1 [level 2], output di WRAP per WW)

I collegamenti sono ordinati per lunghezza calante.

Formato :

A B C D E F G H I K L M N

dove:

- A = numero d'ordine del collegamento
- B = numero del BIN (octal)
- C,D = x , y primo punto (200esimi di pollice)
- E,F = x , y secondo punto ( " " )
- G = lunghezza del filo in centimetri
- H,I = componente , pin primo punto
- K,L = componente , pin secondo punto

M = nome del segnale  
N = numero d'ordine del segnale

Esempio:

1	24	1560	1300	360	280	21.2	U516	18	U104	19	SB1	466
2	21	2600	300	2000	1440	18.2	U127	1	U621	9	SC1	469
3	21	2560	260	2000	1460	18.8	U126	16	U621	8	SB3	468
4	20	1760	780	460	740	17.3	U318	18	U305	20	M31	410
5	20	1760	760	460	720	17.3	U318	17	U305	19	M30	409
6	20	1760	740	460	700	17.3	U318	16	U305	18	M29	408
7	20	1760	720	460	680	17.3	U318	15	U305	17	M28	407
8	20	1760	700	460	660	17.3	U318	14	U305	16	M27	406
9	20	1760	680	460	640	17.3	U318	13	U305	15	M26	405
10	20	1760	660	460	620	17.3	U318	12	U305	14	M25	404
11	20	2700	720	1400	760	17.3	U328	6	U315	4	A05	6
12	20	2700	660	1400	740	17.5	U328	9	U315	5	A04	5
13	20	2760	720	1400	700	18.0	U328	15	U315	7	A02	3
14	17	1760	140	1560	1280	16.1	U118	10	U516	17	SB2	467
15	17	1560	960	360	800	16.7	U416	14	U304	23	OMCKA	423

etc., etc. fino all'EOF

File .CHK (Check-file, output di WRAP per WW)

Serve per controllare sulla OK WWM-600 la continuità dei segnali. La wrappatrice si posiziona sul primo e l'ultimo punto di ogni segnale.

Formato : identico a quello dei files .LV1 e .LV2.

Note: il numero del BIN e la lunghezza del filo hanno valori "dummy".

Esempio :

1	0	360	340	2760	800	99.0	U204	13	U328	19	A00	1
2	0	360	360	2760	480	99.0	U204	14	U228	16	A01	2
3	0	360	380	2760	460	99.0	U204	15	U228	15	A02	3
4	0	360	400	2760	400	99.0	U204	16	U228	12	A03	4
5	0	2700	400	360	420	99.0	U228	9	U204	17	A04	5
6	0	360	440	2700	460	99.0	U204	18	U228	6	A05	6
7	0	360	460	2700	480	99.0	U204	19	U228	5	A06	7
8	0	360	480	2700	540	99.0	U204	20	U228	2	A07	8
9	0	260	340	2660	800	99.0	U203	13	U327	19	A08	9
10	0	260	360	2660	480	99.0	U203	14	U227	16	A09	10
11	0	260	380	2660	460	99.0	U203	15	U227	15	A10	11

etc.,etc. fino all'EOF



File .ERR (Messaggi di avvertimento, output di WRAP per WW e MW)

formato:

segnale            compon.    pin        x        y        warning

Esempio:

AP01	U007	16	14200	0	SEGNALE SU UN SOLO PIN
AP02	U007	14	14400	0	SEGNALE SU UN SOLO PIN
AP03	U007	12	14600	0	SEGNALE SU UN SOLO PIN
AP04	U007	10	14800	0	SEGNALE SU UN SOLO PIN
AP05	U007	8	15000	0	SEGNALE SU UN SOLO PIN
AP06	U007	6	15200	0	SEGNALE SU UN SOLO PIN
AP07	U007	4	15400	0	SEGNALE SU UN SOLO PIN
AP08	U007	2	15600	0	SEGNALE SU UN SOLO PIN
CK120L	U005	22	9700	0	SEGNALE SU UN SOLO PIN
CPUEN	U005	30	8900	0	SEGNALE SU UN SOLO PIN

.  
.  
.

etc.,etc. fino all'EOF

File .HLS (Holes-file, output di WRAP per MW, input a MDS2)

Formato : il primo dato di ogni record è il drill\_code; seguono fino a cinque coppie di coordinate x y in millesimi di pollice.

Esempio:

1	3300	100	3200	100	3100	100	3000	100	2900	100
1	2800	100	2700	100	2600	100	2500	100	2400	100
1	2300	100	2200	100	2100	100	2000	100	1900	100
1	1800	100	1700	100	1600	100	1500	100	1400	100
1	1300	100	1200	100	1100	100	1000	100	900	100
1	800	100	700	100	600	100	500	100	400	100
1	300	100	200	100						
1	3300	200	3200	200	3100	200	3000	200	2900	200
1	2800	200	2700	200	2600	200	2500	200	2400	200
1	2300	200	2200	200	2100	200	2000	200	1900	200
1	1800	200	1700	200	1600	200	1500	200	1400	200
1	1300	200	1200	200	1100	200	1000	200	900	200
1	800	200	700	200	600	200	500	200	400	200
1	300	200	200	200						
1	7200	100	7100	100	7000	100	6900	100	6800	100
1	6700	100	6600	100	6500	100	6400	100	6300	100
1	6200	100	6100	100	6000	100	5900	100	5800	100

.  
.  
.

etc.,etc. fino all'EOF

File .MWN (Netlist-file, output di WRAP per MW, input a MDS2)

Formato: il primo dato di ogni record è il numero d'ordine del segnale; seguono fino a cinque coppie di coordinate x y in millesimi di pollice.

Esempio:

```
1 15800 2300      6500 1600      6900 1600      13200 4700      13200 6900
1 13200 9100      13200 11300     8400 11300     8400 9100      8400 6900
1 8400 4700       5200 4700       5200 6900     5200 9100      5200 11300
1 400 11300       400 9100        400 6900       400 4700       200 2300
2 15800 2200     13300 2600     13300 4800     13300 7000     13300 9200
2 8500 9200      8500 7000      8500 4800      8500 2600      6900 1800
2 6500 1800      5300 2600      5300 4800      5300 7000      5300 9200
2 500 9200       500 7000       500 4800       500 2600       200 2200
3 15800 2100     6500 2000      6900 2000     12900 4700     12900 6900
3 12900 9100     12900 11300    8100 11300     8100 9100      8100 6900
3 8100 4700      4900 4700      4900 6900      4900 9100      4900 11300
3 100 11300      100 9100       100 6900       100 4700       200 2100
4 15800 2000     13100 2600     13100 4800     13100 7000     13100 9200
4 8300 9200      8300 7000      8300 4800      8300 2600      6900 2200
4 6500 2200     5100 2600     5100 4800     5100 7000     5100 9200
4 300 9200       300 7000       300 4800       300 2600       200 2000
```

etc.,etc. fino all'EOF

Files .BOR, .OBD, .OBM, .SPC, .SPL (input a MDS2)

Per un'esauriente descrizione di questi files si consulti:

M.F. Letheren CERN - DD "Cern VAX/VMS command procedure for multiwire board layout".

Esempi:

```
C FILE 16MBZ.BOR PER MW 18/11/86
C
GR 1 1
LI
WI 1
AB 0 0 X16000 Y12000 X-16000 Y-12000
FI
```

```
C FILE 16MBZ.OBD PER MW 18/11/86
C
C
GR 1 1
LI
WI 1
AB 50 50 X15900 Y11650 X-250 Y250 X-15400 Y-250 X-250 Y-11650
FI
```

```
C file 16MBZ.OBM PER MW. LINE OBSTRUCTIONS BETWEEN CONNECTORS ROWS
C 18/11/86
C
GR 1 1
LI
WI 1
C
C
AB 200 150 X3100
AB 4100 150 X3100
AB 8800 150 X3100
AB 12700 150 X3100
AB 6000 11550 X3100
AB 6000 11650 X3100
AB 6000 11750 X3100
AB 6000 11850 X3100
FI
```

```
C FILE 16MBZ.SPC per MW 18/11/86
C
C
FD 80
RT 4 9 42 0 0 1 1 0 0 0
BP 10
GT 1 42 2 42 4 0
FI
```

```
C DUMMY FILE 16MBZ.SPL PER MW 18/11/86
C
FI
```

File PCK001.DAT (drill-code, output di MDS2)

Si ricorda che i files PCKxxx.DAT sono files binari: l'esempio seguente è stato ottenuto mediante un programma di lettura (binaria) e scrittura (formattata).

Si noti che le informazioni utili cominciano dal decimo record, che il terzo record contiene nell'ordine:

Nmax\_records, xmin, ymin, xmax, ymax

e la struttura dell'ultimo record (eof logico).

Esempio:

13873	16973	8224	8224	8224
14641	20013	22095	14381	54
3802	1100	-5100	16900	6600
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
3	4300	-5100	1	1
3	4200	-5100	1	1
3	4100	-5100	1	1
3	4000	-5100	1	1
3	3900	-5100	1	1
3	3800	-5100	1	1
3	3700	-5100	1	1
3	3600	-5100	1	1
3	3500	-5100	1	1
3	3400	-5100	1	1
3	3300	-5100	1	1

.  
.  
.  
.  
-1 32767 32767

File PCK101.DAT (wiring-code, output di MDS2)

Valgono le note dell'esempio precedente.

Esempio:

13873	16973	8224	8224	8224
14641	20013	22095	14381	54
4909	1100	-5700	16900	6100
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
5	13100	-5600	169	645
0	13100	-5700	169	645
0	14100	-5700	169	645
6	14100	-5600	169	645
5	12900	-5300	674	640
6	12900	-5600	674	640
5	13800	-3600	686	573
0	13900	-3600	686	573
0	13900	-3800	686	573
6	13800	-3800	686	573
5	12600	-3300	93	558
6	12700	-3400	93	558

.  
.  
.  
-1 32767 32767 (eof logico)

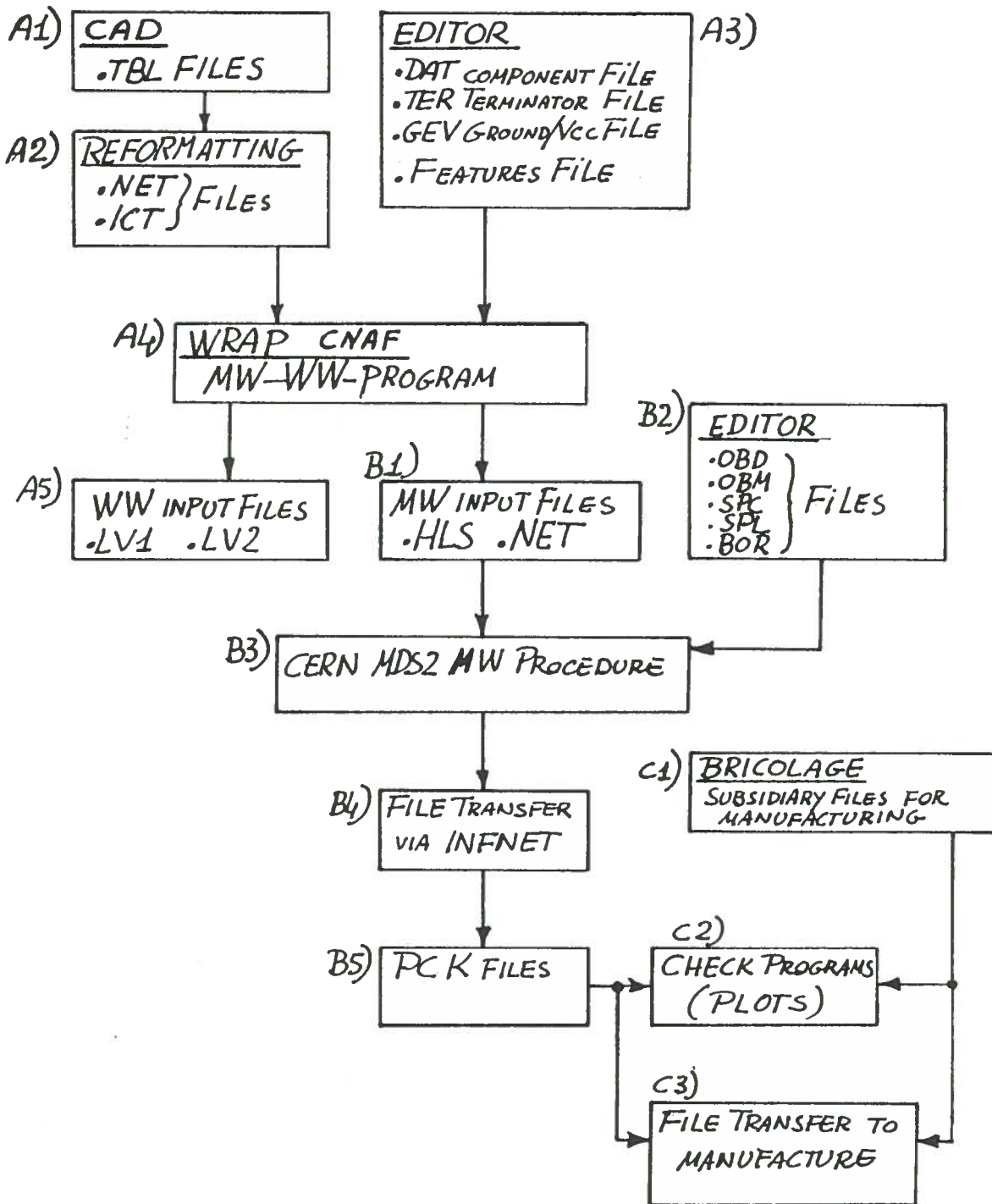


FIG. 1