

DEUTSCHES ELEKTRONEN-SYNCHROTRON **DESY**

DESY 73/54
November 1973

DESY-Bibliothek

1. FEB. 1974

MEDEA, ein Spezialrechner zur Vorverarbeitung und Formatierung
von Proportionalkammer-Daten

von

H.-J. Stuckenberg

2 HAMBURG 52 · NOTKESTIEG 1

MEDEA, ein Spezialrechner zur Vorverarbeitung und Formatierung
von Proportionalkammer-Daten

von

H.-J. Stuckenberg

Abstract

MEDEA is a special computer for handling and formatting data from proportional chambers. It converts incoming data from a bit-to-address-readout system into binary addresses. A built-in RePROM contains the number of counting wires per chamber plane used during the experiment. The address of the last wire of plane (n-1) will be subtracted from the address under surveillance in plane n; this address is the wire number in plane n; it is compared with the address of the previous operation for checking whether it is a single or a cluster address. The binary address is then added to the cluster length divided by two to obtain the center address of a cluster. The result is loaded into a buffer memory, from where it will be transmitted to a PDP-11 computer.

1. Einleitung

MEDEA ist ein Spezialrechner zur Verarbeitung und Formatierung von Drahtkammerdaten mit anschließender Speicherung. MEDEA steht für

Memory for
Experiment
Data in
Event
Access.

Zweck der Anlage ist es, Daten aus Proportional- und Funkenkammern, speziell aus CATO-Karten [1] so aufzubereiten, daß die Adressen der gesetzten Drähte als binäre Zahlen innerhalb der zugehörigen Drahtebene (von Drahtnummer Null bis zum Ebenenrand gerechnet) gebildet werden.

Die vom PERSEUS-Auslesesystem [2] ankommenden Adressen, deren Bits, wie Bild 1 zeigt, aus Crate-Nummern, Karten-Nummern innerhalb des Crates und Draht-Nummern innerhalb der Karte bestehen, müssen in eine monotone Folge von Binärzahlen umgerechnet werden. In einem wiederprogrammierbaren Speicher, einem RePROM, sind die Ebenenlängen, d.h. die Anzahl der Drähte jeder Ebenen, abgelegt. Mit ihnen werden die binären Drahtadressen verglichen und festgestellt, ob die gefundenen Adressen innerhalb der n-ten Ebene ist, falls nicht, wird aus dem RePROM die Ebenenlänge $n + 1$ geholt und der Vergleich erneuert. Dies wird so oft ausgeführt, bis die zugehörige Ebene gefunden und, durch Subtraktion der Binäradresse von der Endadresse der vorangehenden Ebene, die Draht-Nummer innerhalb der Ebene festgelegt ist.

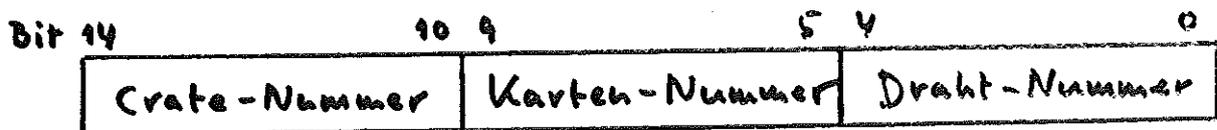


Bild 1

Dann wird durch Vergleich mit der Vorgängeradresse (diese ist zwischengespeichert, der Rechner ist immer eine Adresse im Rückstand) geprüft, ob die neue Adresse einen Einzeldraht oder einen Draht in einem Cluster darstellt. Im ersten Fall wird die Adresse in einem Register abgelegt, im zweiten Fall der Clusterzähler inkrementiert.

Aus Drahtadresse und Clusterlänge (bei Einzeldrähten gleich 1) wird dann durch Addition die mittlere Drahtadresse gebildet und zwar auf Wunsch der Benutzer der doppelte Wert, d.h. $2 \times$ mittlere Adresse = $2 \times$ Anfangsadresse + Clusterlänge. Dieser Wert wird mit der zugehörigen Ebenen-Nummer sowie der Clusterlänge in 2 Speicherplätzen eines 1K-Worte \times 16 Bit-Speichers abgelegt, d.h. jede akzeptierte Adresse steht wie folgt in den beiden Worten (Bild 2).

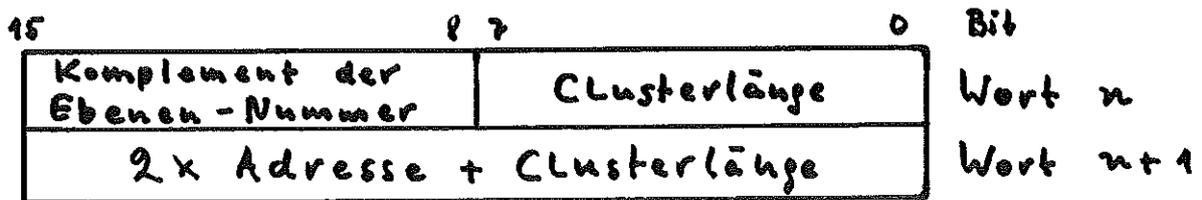


Bild 2

Alle abgespeicherten Worte mit Adressen einer Ebene werden blockweise an die PDP übertragen.

Hat das PERSEUS-System alle Adressen bearbeitet, sendet es ein End of Record (EOR)-Signal an den MEDEA-Rechner. Dieser informiert die PDP und überträgt abschließend so viele Pointerworte, wie Ebenen angeschlossen sind. Der Inhalt eines Pointerwortes gibt die Speicheradresse an, in der innerhalb des gesamten Datenblocks die Adressen der Ebene beginnen, der der Pointer zugeordnet ist.

Die Pointer werden den Adressen im Block vorgestellt, als erstes Wort wird schließlich der Inhalt eines im Eingangskanal der PDP mitlaufenden Word-Count-Registers gesetzt.

Der Verkehr zwischen PERSEUS und MEDEA einerseits bzw. MEDEA und der PDP andererseits wird asynchron nach einem Handshakeverfahren abgewickelt. Auf ein START-Signal wird jede gefundene Adresse durch ein ADF (Adress found)-Signal angeboten und durch das LONA (Look for new address)-Signal quittiert. Das EOR-Signal beschließt den Verkehr.

Die Zeit zur Umrechnung und Speicherung einer Adresse dauert normalerweise 1.6 μ sec.

2. Beschreibung der Logik-Blöcke

Die Gesamtschaltung, die Bild 3 zeigt, ist in 8 Logik-Blöcke aufgeteilt, die auf je einer Karte realisiert sind.

Alle Logik-Blöcke werden von einem Taktsystem gesteuert, das jede 100 nsec eine Operation ausführt. Die Takte werden mit T_0 , T_1 ... bezeichnet. Der vollständige Ablauf der Adressenbehandlung ist im Flußdiagramm beschrieben, das sich im Anhang befindet.

2.1 Adressen-Umrechnung

Ein Experiment mit maximal 20 000 Drähten benötigt 32 Crates, um die CATO-Karten unterzubringen, denn jede Karte enthält die Elektronik für 32 Drähte und 20 Karten faßt das Crate, so daß die Signale aus 640 Drähten in einem Crate verarbeitet werden können. Diese Konfiguration der Bits ist in dem Datenwort enthalten, das das PERSEUS-Auslesesystem abgibt und das in Bild 1 dargestellt war.

Daraus läßt sich relativ leicht die Binäradresse BA errechnen, es ist

$$BA = D + 32K + 640 C$$

wenn

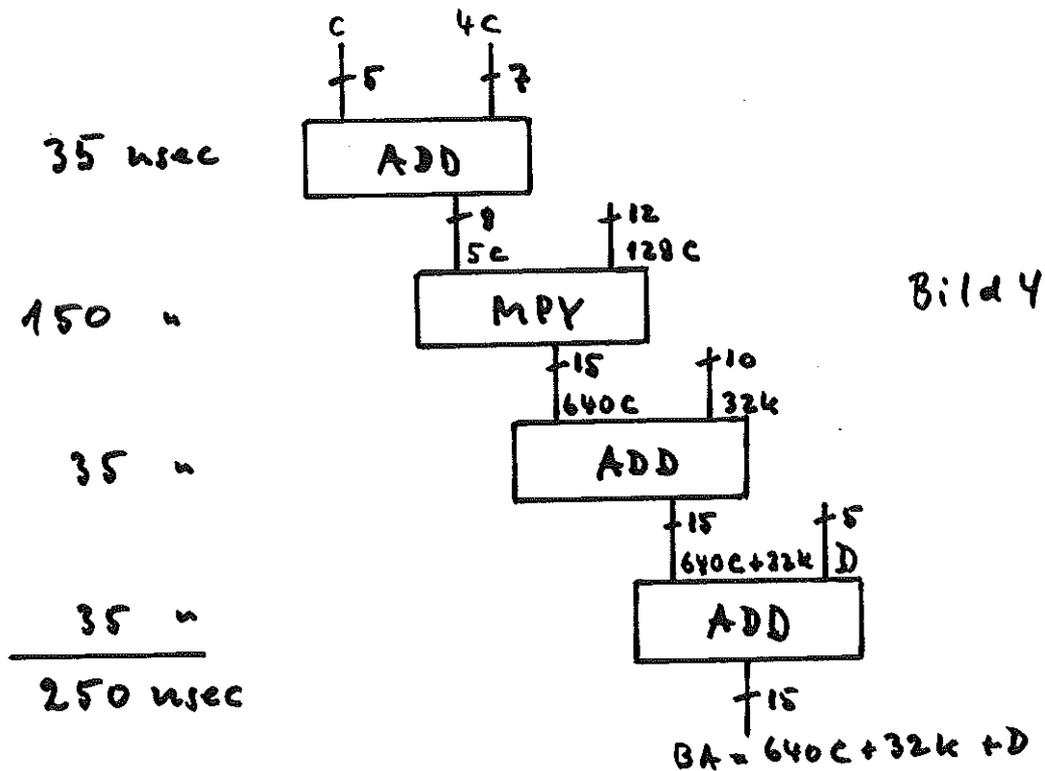
D = Draht - Nummer

K = Karten - Nummer

C = Crate - Nummer ist.

Da 32 eine echte Binärzahl ist, kann man sie durch feste "Stellenverschiebung" um 5 Stellen verdrahten, ähnlich kann man 640 in $(4+1) \cdot 128$ aufspalten und aus der Addition von $4C$ (2 Stellen verschoben verdrahtet) und C dann $5C$ erhalten, das mit 128 (um 7 Stellen verschoben verdrahtet) multipliziert wird. Schließlich kann die Draht-Nummer direkt addiert werden. Das Schaltbild sieht also wie folgt aus (Bild 4, links stehen die zur Arithmetik benötigten Zeiten).

Die PERSEUS-Adressen werden also mit Takt 1 des unter 2.5 beschriebenen Clock-generators nach dem ADF von PERSEUS in das Eingangslatch übernommen, die Adressen-Umrechnung startet und kann bis zum Ende von Takt 4 dauern, das Ergebnis wird mit Takt 5 in Datenlatch 1 geladen.



2.2 Bildung der Adressen-Nummer in der Ebene

Im RePROM (vgl. 2.8) liegen zu jeder Ebenen-Nummer die Zahl der in der Ebene vorhandenen Drähte, kurz Ebenenlänge EL genannt. Die erste Ebenenlänge wird zu Beginn mit Takt \emptyset in den Ebenenlänge-Komparator-Latch geladen. Die Binäradresse BA im Datenlatch 1 (vgl. 2.1) wird mit Takt 6 im Ebenenlängen-Komparator verglichen und festgestellt, ob $BA \leq EL$ oder $BA > EL$ ist.

Ist $BA \leq EL$, wird mit Takt 8 der IN-Ebenen-FF gesetzt und die BA in das Minuendenregister geladen. Im Subtrahendenregister steht die Endadresse der vorangehenden Ebene (falls in Ebene 1 kompariert wird, ist die Endadresse Null), mit dieser wird die Subtraktion $BA - EL$ ausgeführt und die Differenz mit der Rückflanke des Taktes 7 in das Datenlatch 2 geladen.

Da die Differenz zur Endadresse der vorangehenden Ebene (n-1) gebildet wird, ist die Binäradresse in der Ebene n gefunden und zwar als Zahl zwischen Null und Endadresse der n-ten Ebene. Diese Differenzbildung setzt voraus, daß im RePROM die aufsummierten Endadressen stehen, d.h. allgemein ergibt sich die Differenz aus BA (Ebene n) - $\sum_{i=1}^{n-1} EL_i$.

Ist $BA > EL$, wird der OUT-Ebenen-FF gesetzt. Falls in dieser Ebene schon Adressen gefunden wurden, werden diese an die PDP übertragen. Dies geschieht durch Umschalten des WC-Registers von UP auf DOWN sowie durch die ADF- und LONA-Signale, die für jedes Wort ausgesendet werden. Dieser Vorgang ist in Abschnitt 2.5 beschrieben.

Sind keine Adressen gefunden oder sind sie schon an die PDP übertragen, ist der Inhalt des WC-Registers Null. Dann wird Clock 2 gestartet, deren 3 Takte (T21 bis T23) zunächst den Inhalt des Pointerzählers (vgl. Abschnitt 2.4) auf die durch die Ebenen-Nummer festgelegte Speicheradresse des 64 Worte x 16 Bit-Speichers MEM2 laden, das Subtrahendenregister erst clearen und dann mit dem Inhalt des Ebenen-Komparatorlatch (Endadresse der vorigen Ebene) laden. Mit dem dritten Takt wird die Adresse des RePROM inkrementiert, d.h. eine neue Ebene angewählt und die zu dieser Adresse gehörende neue Ebenenlänge in den Ebenen-Komparatorlatch gegatet. Mit dem gleichen Takt wird der OUT-Ebenen-FF gecleart.

Dieser Vorgang des Aufrufens neuer Ebenen wird so oft wiederholt, bis $BA \leq EL$ ist. Falls in einer Ebene keine Adresse gefunden wurde, ist der Inhalt des Pointerwortes identisch mit dem seines Vorgängers.

Nachdem $BA \leq EL$ ist, läuft die Differenzbildung wie oben beschrieben ab, bis die Differenz in Datenlatch 2 steht.

Mit Takt 8 wird der Inhalt des Datenlatch 2 einmal nach links verschoben, um die gewünschte Verdoppelung der Binäradresse zu erhalten (vgl. Wortformat nach Bild 2).

2.3 Einzeldraht oder Clusterdraht?

Ob die bearbeitete Adresse eine Einzel- oder eine Clusterdrahtadresse ist, kann erst entschieden werden, wenn im nächsten Zyklus die folgende Adresse behandelt wird. Dann wird sie mit der um 1 erhöhten Vorgängeradresse verglichen, die mit Takt 16 des vorigen Zyklus ins Vorgängerlatch geladen wurde.

Der Vergleich der alten Adresse mit der neuen wird mit Takt 9 vorgenommen, der den Clusterlängen-Komparator frei gibt und die in Datenlatch 2 stehende Adresse auf den Komparator schaltet. Das Ergebnis der Vergleichsoperation zeigt, ob die Binäradresse BA gleich der Vorgängeradresse VA +1 ist oder nicht.

Ist $BA = VA + 1$, wird der Clusterlängen-FF gesetzt und damit der Clusterzähler inkrementiert. Mit Takt 16 wird der Inhalt des Datenlatch 2 (vgl. 2.2) in den Vorgängerlatch geladen, dabei um 1 erhöht. Mit der Vorderflanke des Takt 16 wird auch das in diesem Fall nicht benutzte Σ -Register gecleart, mit der Rückflanke der Clusterlängen-FF.

Ist $BA \neq VA + 1$, wird der Clusterlängen-FF nicht gesetzt, da es sich um eine Einzeldrahtadresse handelt. Die weitere Behandlung der Adresse muß 2 Möglichkeiten unterscheiden:

- es ist die erste Binäradresse, die vom PERSEUS-System angeliefert wurde,
- im ADD2-Latch liegt bereits eine Binäradresse, die entweder aus dem vorigen Zyklus stammt und sich in diesem Zyklus als Einzeldrahtadresse erweist ($BA \neq VA + 1$) oder aus vorigen Zyklen kommt und die Anfangsadresse eines Cluster ist. Im letzten Fall zeigt das in diesem Zyklus gewonnene Ergebnis $BA \neq VA + 1$, daß die Clusterdrahtadressen beendet sind und eine neue Adresse vorhanden ist, die sich entweder wieder als Einzeladresse oder als Beginn eines neuen Clusters herausstellen wird.

Der ersten Möglichkeit entsprechend wird mit Takt 16 die Binäradresse ins Vorgänger- und ADD2-Latch geladen und das hier nicht benutzte Σ -Register und der Clusterlängen-FF gecleart.

Die zweite Möglichkeit bedeutet, daß, gleichgültig, ob im ADD2-Latch eine Einzeldrahtadresse oder die Anfangsadresse eines Clusters steht, diese Binäradresse mit dem Inhalt des Clusterzählers addiert und in den Speicher MEM1 (vgl. 2.6, 2.7) übertragen werden muß.

Zuvor aber muß, Bild 2 entsprechend, das Komplement der Ebenen-Nummer sowie der Inhalt des Clusterzählers mit Takt 10' in das HI- bzw. LO-BYTE-Register geladen werden und mit Takt 12' in den Speicher MEM1 transportiert werden. Die zugehörige decodierte Adresse liefert der Adressenzähler, der in 2.5 beschrieben wird.

Zur Addition wird mit Takt 13' der Inhalt des Clusterzählers in das ADD1-Latch übertragen, die arithmetische Operation ausgeführt und das Ergebnis ins Σ -Register geladen. Wieder liefert der schon erwähnte Adressenzähler eine decodierte Speicheradresse, so daß mit Takt 15' der Inhalt des Σ -Registers, also $2 \times$ Binäradresse + Clusterlänge in den Speicher geschickt wird. Gleichzeitig wird der Clusterzähler gecleart.

Die Takte $T10'$.. $T15'$ werden aus den Takten $T10$.. $T15$ erzeugt, wenn gleichzeitig ein LOAD-Signal vorhanden ist, das im ersten Zyklus durch Takt 16 gesetzt wird und anzeigt, daß der ADD2-Latch mit einer Adresse geladen ist. Außerdem dürfen die Takte $T10'$.. $T15'$ nur arbeiten, wenn $BA \neq VA + 1$, da in dem anderen Fall nur der Clusterzähler inkrementiert werden soll, aber keine Addition ausgeführt werden darf. Die Takte bedeuten also $T10' .. T15' = (T10 .. T15) \cdot \text{LOAD} \cdot \neq$ -Funktion.

Nach der beschriebenen Behandlung der im ADD2-Latch stehenden Binäradresse aus dem vorigen Zyklus (oder den vorigen Zyklen) wird mit Takt 16 der Inhalt des Datenlatch 2, also die neue Binäradresse, sowohl in den ADD2-Latch als auch in den Vorgängerlatch geladen, das Σ -Register und der Clusterlängen-FF gecleart und die neue, nun im ADD2-Latch stehende Adresse wartet auf die Ankunft des nächsten, vom PERSEUS-System kommenden Adressenwortes, mit dem ein neuer Zyklus gestartet wird.

Ist am Ende der Verarbeitung der Drahtadressen aus einem Event der EOR(PERS)-FF gesetzt, steht im ADD2-Register noch die letzte Adresse, die mit dem Inhalt des Clusterzählers addiert und in den Speicher MEM1 transportiert werden muß, in dem die Adressen der letzten Drahtebene stehen.

Dazu wird Clock 3 gestartet (vgl. Abschnitt 2.5), mit Takt T31 sowohl das HI-LO-BYTE-Register geladen als auch die Addition ausgeführt mit den Takten T32 und T33, die Adresse decodiert und der Inhalt des HI-LO-Register ins MEM1 geladen, mit den Takten T34 und T35 geschieht das Gleiche für den Inhalt des Σ -Registers. Anschließend muß der Speicherinhalt, also die gesetzten Adressen der letzten Ebene an die PDP transportiert werden, wie es in Abschnitt 2.5 beschrieben ist.

2.4 Erzeugung der Pointerworte

Das Pointerwort n , das der Ebene n zugeordnet ist, enthält die Speicheradresse, in der die zu seiner Ebene gehörenden Daten im Speicherblock stehen.

Da mit jedem Takt $T10'$ und $T13'$ ein Wort in den Speicher transportiert wird, muß auch der Inhalt des Pointerzählers mit jedem dieser Takte inkrementiert werden.

Beim Umschalten des RePROM (vgl. Abschnitt 2.2 und 2.8) auf eine neue Ebene wird der gerade anstehende Inhalt des Pointerzählers in den Speicher MEM2 transportiert, der 64 Worte je 16 Bit speichern kann. Diese Größe entspricht

der bei den DASP-Experimenten eingesetzten Zahl von Proportionalkammern. Da aber auch weniger als 64 Proportionalkammer-Ebenen benutzt werden können, wird der als Adresszähler arbeitende Ebenenzähler mit dem Komplement der tatsächlich verwendeten Ebenen zu 64 vorgeladen.

Beim Umschalten der Ebenen-Nummer wird mit Takt T21 der Inhalt des Ebenenzählers decodiert, mit Takt T22 der Inhalt des Pointerzählers in den Speicher geladen. Takt T22 inkrementiert außerdem den Ebenenzähler.

Mit Erscheinen des EOR(PERS)-Signals muß der Inhalt des Ebenenzählers noch um 1 inkrementiert werden, um den Ebenen OVF-FF zu setzen. Hier müssen zwei Fälle unterschieden werden, nämlich

- die letzt gefundene Adresse war wirklich in der letzten Ebene (Nummer 64) oder
- in der letzten Ebene wurde kein Draht gesetzt, d.h. es gibt auch keine gefundene Adresse in Ebene 64. Dann muß der Ebenenzähler weiter inkrementiert werden, bis der OVF-FF gesetzt wird, damit auch die Pointer der letzten Ebene erzeugt werden.

Die Operationen für diese beiden Fälle laufen wie folgt ab:

Fall 1: der Ebenenzähler ist voll (d.h. 111 111). Takt T33 setzt den Inkrement-FF, die Worte der letzten Ebene werden zur PDP übertragen, nach dem letzten Wort wird der WC-OVF-FF gecleart. Dieser Übergang inkrementiert den Ebenenzähler, der Ebenen-OVF-FF wird gesetzt.

In der Zwischenzeit wurde der Start-Clock4-FF gesetzt und zwar etwa 75 nsec nach dem Inkrementpuls. Das \bar{Q} -Signal des Ebenen-OVF-FF liefert nach etwa 65 nsec den Pegel LOW an das Clock4-Gate, so daß kein weiterer Takt in den Zähler gelangt.

Das Setzen des Ebenen-OVF-FF liefert mit dem EOR(PERS)-Signal eine UND-Bedingung zum Setzen des EOR(PDP)-FF. Dieses Q-Signal geht an die PDP als EOR-Signal (vgl. Abschnitt 2.5). Mit dem Setzen des EOR(PDP)-FF wird der Start-Clock4-FF wieder gecleart.

Fall 2: der Ebenenzähler ist nicht voll.

Takt T33 setzt den Inkrement-FF, die Worte der letzten aufgerufenen Ebene werden zur PDP übertragen, nach dem letzten Wort wird der WC-OVF-FF gesetzt und damit

der Inkrement-FF gecleart. Dieser Übergang inkrementiert den Ebenenzähler, der Ebenen-OVF-FF wird aber noch nicht gesetzt.

In der Zwischenzeit wurde der Start-Clock4-FF gesetzt und zwar etwa 75 nsec nach dem Inkrementpuls. Das \bar{Q} -Signal des Ebenen-OVF-FF bleibt auf dem Pegel HIGH, damit wird das Clock4-Gate geöffnet und Clock 4 startet mit Pulsen im 100 nsec-Abstand. Die Pulse inkrementieren den Inhalt des Ebenenzählers, bis der Ebenen-OVF-FF gesetzt wird. Dann erfolgt die weitere Behandlung wie unter Fall 1.

Jeder Clock 4-Puls gibt nach 50 nsec Delay den Inhalt des Ebenenzählers zum Decodieren frei, nach weiteren 50 nsec wird der Inhalt des Pointerzählers in den Speicher MEM2 geladen.

2.5 Clock- und Zähllogik, Kontrollsignale

Vom Experiment kommt das Startsignal, das über einen NIM-TTL-Konverter den Start-FF setzt. Das Q-Signal dieses FF wird wieder auf NIM-Pegel konvertiert und als BUSY-Signal ausgegeben.

Die Vorderflanke des Q-Signals triggert den Univibrator OS1, der zwei Startsignale erzeugt, nämlich

- den Start an die PERSEUS-Einheit und
- den General Clear, mit dem die FF und Register auf Null gestellt werden.

Außerdem triggert der erste Univibrator einen zweiten (OS2), dessen Ausgangssignal als Takt \emptyset dient sowie den Clock-FF setzt und damit die 10 MHz-Clock für die Erzeugung der ersten 16 Takte bereitstellt.

Trifft vom PERSEUS-System nach Auffinden der ersten gesetzten Drahtadresse das ADF-Signal ein, wird der ADF(PERS)-FF gesetzt und die Clock 1 zählt in den 4 Bit-Zähler die Takte ein. Takt 16 cleart den ADF(PERS)-FF und setzt den LONA-FF dessen Q-Ausgang das LONA-Signal darstellt und an das PERSEUS-System übertragen wird.

Da mit jedem Takt T10' und T13' ein Wort in den Speicher MEM1 geladen wird, müssen sowohl der Adressenzähler als auch das durch den General Clear auf UP gestellte Ebenen-WC-Register mit jedem dieser Takte inkrementiert werden.

Ist die erzeugte Binäradresse größer als die z.Zt. im Ebenenlängen-Komparator stehende Ebenenlänge, d.h. ist $BA > EL$, wird der Clock-FF gecleart und damit Clock 1 gestoppt.

Ist dann der Inhalt des Ebenen-WC-Registers $\neq 0$, d.h. ist in der z.Zt. bearbeiteten Ebene mindestens eine Adresse gefunden, wird Clock 2 noch nicht freigegeben, sondern über den Univibrator OS3 das Ebenen-WC-Register auf DOWN geschaltet und, nach einer Verzögerung von 500 nsec, der OUTPUT-FF und damit das ADF-Signal an die PDP gegeben. Gleichfalls wird über das Ausgangsgate der Inhalt des geclearten Adressenzählers an den MEM1-Decoder geschaltet, dort decodiert und das erste Speicherwort im Speicher-Datenlatch bereitgestellt.

Ist dieses Wort von der PDP angenommen, wird es mit dem LONA-Signal quittiert, das den OUTPUT-FF cleart, den Adressenzähler inkrementiert, das Ebenen-WC-Register dekrementiert und über ein Delay den OUTPUT-FF erneut setzt. Damit geht das zweite ADF-Signal an die PDP usw.

Ist das letzte Wort des Speichers MEM1 an die PDP übertragen, ist der Inhalt des Ebenen-WC-Registers = 0, d.h. der WC-OVF-FF wird gesetzt, der den U/D-FF auf UP setzt, den Adressenzähler cleart und sich selbst nach 100 nsec zurückstellt. Das letzte LONA-Signal cleart den OUTPUT-FF.

Da nun das Ebenen-WC-Register leer ist und $BA > EL$, startet Clock 2 und erzeugt die Ebenen-Umschalttakte T21 bis T23, deren Funktion in Abschnitt 2.2 beschrieben ist. Ist $BA < EL$, wird Clock 2 wieder gestoppt, der Clock-FF erneut gesetzt und Clock 1 kann dort weiterzählen, wo sie aufgehört hat.

Mit jedem Takt T11' und T14' wird der Inhalt des Adressenzählers an den MEM1-Decoder gegatet und dort decodiert, um die Speicherworte aus dem HI-LO-BYTE-Register und aus dem Σ -Register in den Speicher MEM1 zu transportieren.

Sind alle gesetzten Drahtadressen vom PERSEUS-System gefunden, wird das EOR(PERS)-Signal angenommen und damit der EOR(PERS)-FF gesetzt.

Dieser Übergang sperrt den Clock-FF und Clock 3 kann starten, um mit den Takten T31 bis T35 die zuletzt gefundene Binäradresse mit dem Inhalt des Clusterzählers zu addieren und das Ergebnis mit dem zugehörigen Inhalt des HI-LO-BYTE-Registers in den Speicher MEM1 zu übertragen, Takt T35 stoppt Clock 3.

Das Q-Signal des EOR(PERS)-FF setzt, nachdem der Inhalt der letzten Ebene an die PDP transportiert ist, zusammen mit den Q-Signalen der gesetzten FF vom Ebenen-OVF und vom UP/DOWN den EOR(PDP)-FF. Das Q-Signal dieses FF liefert das EOR-Signal an die PDP und zeigt damit an, daß alle gefundenen und neu berechneten Drahtadressen an die PDP abgeliefert wurden.

Nun fehlt noch die Übertragung der Pointerworte. Nach Erscheinen des Q-Signals aus dem EOR(PDP)-FF wird der OUTPUT-FF gesetzt, das ADF an die PDP geht auf HIGH und die 64 Speicherworte aus MEM2 werden mit ADF- und LONA-Signalen an die PDP transportiert. Dazu wird das LONA-Signal mit dem Q-Signal des EOR(PDP)-FF gegatet, da sonst der MEM2-Inhalt bereits früher zusammen mit dem Inhalt von MEM1 geleert würde. Der EOR(PDP)-FF wird mit dem Overflow des 6 Bit-Zählers zurückgestellt, der die Pointer-LONA-Signale mitzählt.

Damit ist die gesamte Operation für ein Event abgeschlossen.

Sind im Event 250 Drähte gesetzt, dauert die Verarbeitung in MEDEA	$250 \times 1.6 = 400 \text{ } \mu\text{sec}$
durch Cluster auf 100 Funkenmitten reduziert, ergibt 100 Doppelworte an die PDP	$200 \times 1 = 200 \text{ ''}$
die Übertragung der 64 Pointerworte dauert	$64 \times 1 = 64 \text{ ''}$
die Ebenenumschaltung beansprucht	$64 \times 0.3 = 19.2''$
die Behandlung des letzten Wortes	$5 \times 0.1 = 0.5''$
Der Gesamtablauf dauert also in diesem Beispiel	<u>683.7 μsec.</u>

2.6 Speicher MEM1

Der Speicher MEM1 ist auf 1K Worte je 16 Bit ausgelegt, in ihm werden in steigender Adressenreihenfolge die Doppelworte abgelegt, die eine gefundene und arithmetisch behandelte Drahtadresse beschreiben. Das Format der Worte zeigte Bild 2. Der Speicherkapazität entsprechend können also bis zu 512 Drahtadressen pro Ebene behandelt werden.

Der Speicher ist industriell gefertigt, z.B. Intel Typ IN-52 mit 1K Worten je 8 Bit auf einer Karte und einer Zykluszeit von 150 nsec. Auf der Karte befinden sich außer dem Speicher das Adress- und das Datenregister sowie ein Modul-Selekt-Eingang. Die Schaltungen sind TTL-Kompatibel, die Ein- und Ausgänge sind gepuffert.

2.7 Speicher MEM1

Auf einer Speicherkarte befinden sich 1K Worte je 8 Bit, der Speicher wird also auf 2 Karten verteilt. Da die Abschnitts-Nummern sich auf die Karten-Nummern beziehen, wurde ein Extraabschnitt für die zweite Karte geschrieben.

2.8 Wiederprogrammierbarer Speicher RePROM

Im RePROM liegen die Ebenenlängen, d.h. die Zahl der in jeder Ebene vorhandenen Drähte. Aus Gründen, die in Abschnitt 2.2 beschrieben sind, müssen die summierten Ebenenlängen ins RePROM eingespeichert werden.

Die Eingabe erfolgt mit Schaltern, die Summierung wird durch ein eingebautes Addierwerk ausgeführt. Mit Takt $T\emptyset$ wird das unter der Adresse \emptyset (Inhalt des Ebenenzählers nach dem Start) liegende Wort des RePROM, also die Ebenenlänge der ersten Ebene in den Ebenenlängen-Komparator (vgl. Abschnitt 2.2) geladen, mit Takt T23 jede neue Ebenenlänge.

Anhang

Der in den Abschnitten 2.1 bis 2.8 beschriebene logische und zeitliche Ablauf wird in einem Flußdiagramm beschrieben, das im letzten Bild 5 dargestellt ist.

Literatur

- [1] W. Neff, H.-J. Stuckenberg, G. Will; CATO - Ein Verstärker-Logik-System für Charpak-Kammern mit 32 Kanälen auf einer CAMAC-Karte, Interner Bericht DESY F56-72/2, November 1972.
- [2] H. Brechtel, H.-J. Stuckenberg; PERSEUS - Ein neues, sehr schnelles Auslesesystem für Charpak-Kammern und Hodoskope, Interner Bericht DESY F56-72/1, Oktober 1972.

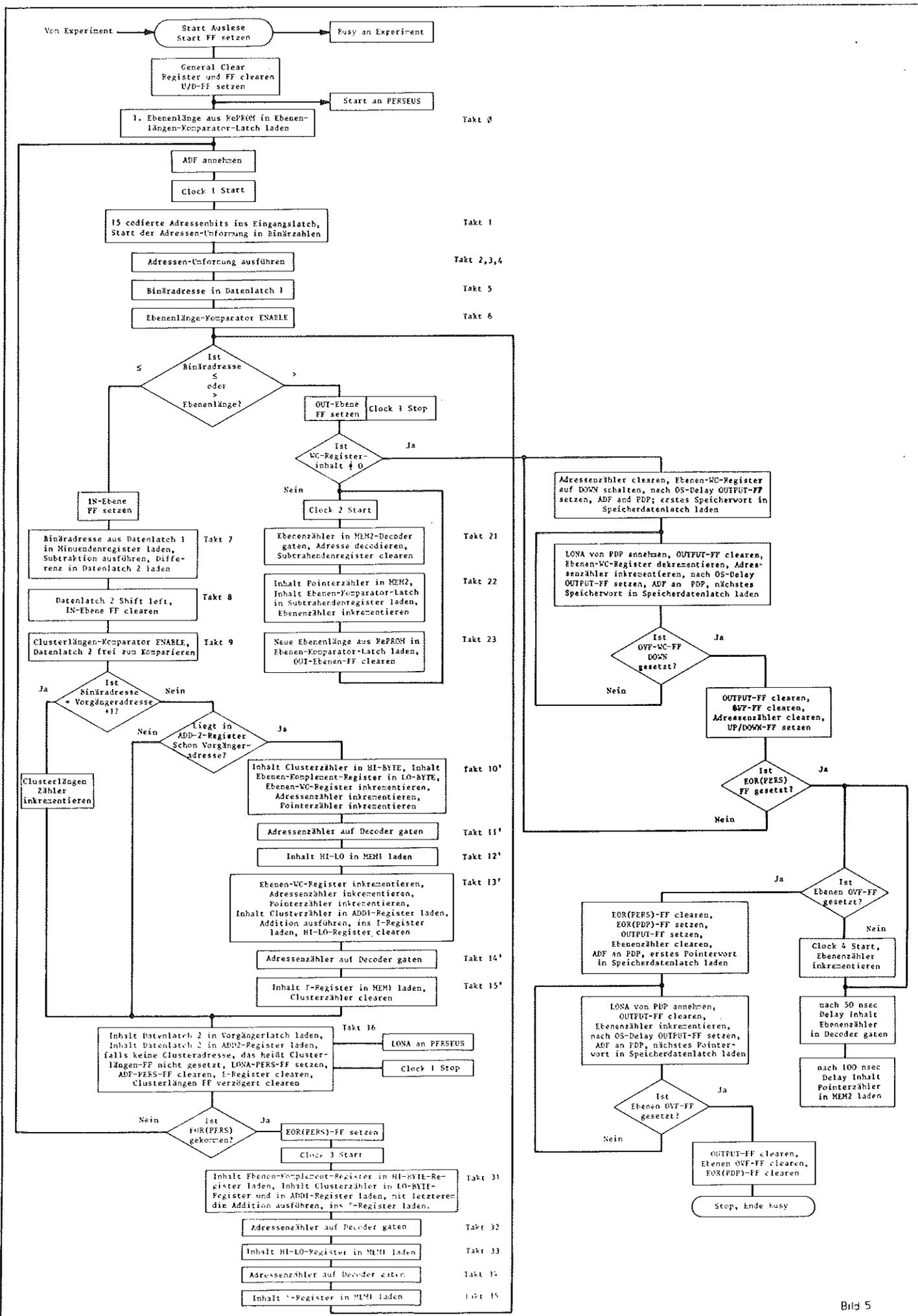


Bild 5