

ファームウェア方式による 汎用映像入出力装置の制御

池田 克夫 森本 潤一
(京都大学 工学部)

清野 武

1 はじめに

われわれは知能端末の研究の一環として、計算機に総合的な視力を与える研究を行なっている。この研究では動的な映像の入出力を行なうためにイメージ・ディスプレイ方式のビデオ・カメラとライトペン付きのCRTグラフィック・ディスプレイを結合した汎用映像入出力装置を構成している。

本汎用映像入出力装置の制御に関しては;

- 1 ビデオ信号を取扱うために超高速の処理を必要とする。
- 2 映像の入出力および映像を介したマン・マシン・インターフェースの制御に関しては今後研究すべき問題が多数あるものと考えられ、制御方式も“柔軟”なものとしたい。

という2点から、プログラム制御方式とし、さらに各命令をファームウェア方式の制御装置により解釈実行することとした。

最近LSIの進歩により1チップCPUが各種の制御装置の中で用いられつつあるが、これらは皆、制御方式の柔軟性・可変性を意図したものであるといえよう。ただ、ビデオ信号の処理が可能な速度を持つものは現在のところ市販されて居ないので、われわれは自作により装置を構成した。

本装置は中央処理装置から起動された後は、中央処理装置の主記憶に格納された制御プログラムを自律的に解釈して実行する。本装置は中央処理装置に割込み信号を送信することができる。この場合は動作を停止する。この方式によってシステムの能率向上とプログラム制御による柔軟性を得ている。

本制御装置のアーキテクチャには方式的にみて次に述べる幾つの特徴がある。

- 1 概念的には、制御装置の外側というものは存在せず、演算や制御は“制御装置の”レジスタまたはフリップ・フロップを対象として行なわれる。
- 2 命令形式を4番地形式として、処理時間の短縮を図っている。
- 3 制御記憶のアドレス方式を“連想方式”としたために、条件の判定、待合セループ、多分岐の実現、サブルーチンの結合と復帰、ROMに実装されたプログラムの修正などが容易、記憶容量も少ないという点で大きな利益があり、制御速度の高速化も図れる。
- 4 この反面、マイクロ・プログラムのステップのフローを追うことがやや困難でプログラムのデバッグに多少問題がある。

現在のところ本制御装置はマシン・サイクル150nsにて命令を実行しており、試験用に作成したプログラムにより、実時間で、256×256絵素(1点4bitの濃淡)の画面をカメラから読取ってCRTに表示している。

2 映像入出力装置の構成

視覚は情報処理という観点からみると人間の五感の内でも最も重要なものである。このような理由から、視覚を用いたマン・マシン・インターフェースの構成が要求される。

計算機の持つ視力は次のような条件を満たすことが要求されるであろう。

- 1 一定範囲の領域を動的な映像としてとらえること。
- 2 ある点または範囲を観測して生じた変化を通知すること。
- 3 動く点や映像を追跡すること。
- 4 目的に応じた制御が可能であること。

これらの要求を満たすためには；

- 1 視線を任意の方向に向けることができること。
- 2 指定された視点で輝度の測定および表示を行なうことができること。

が必要である。

われわれはランダム・スキャン可能な“眼”として、近年その特性が著るしく改善されたイメージ・ディセクタを用い、これにライトペン付きのCRTグラフィック・ディスプレイを結合した汎用映像入出力装置を構成した。

本装置は、フライイング・スポット・スキャナに必要なフィルム化の手段を省いている点で一步進んだランダム・スキャン可能な映像入出力装置である。

本装置のブロック図を図2.1に示す。機能的には次の4つの部分から成っている。

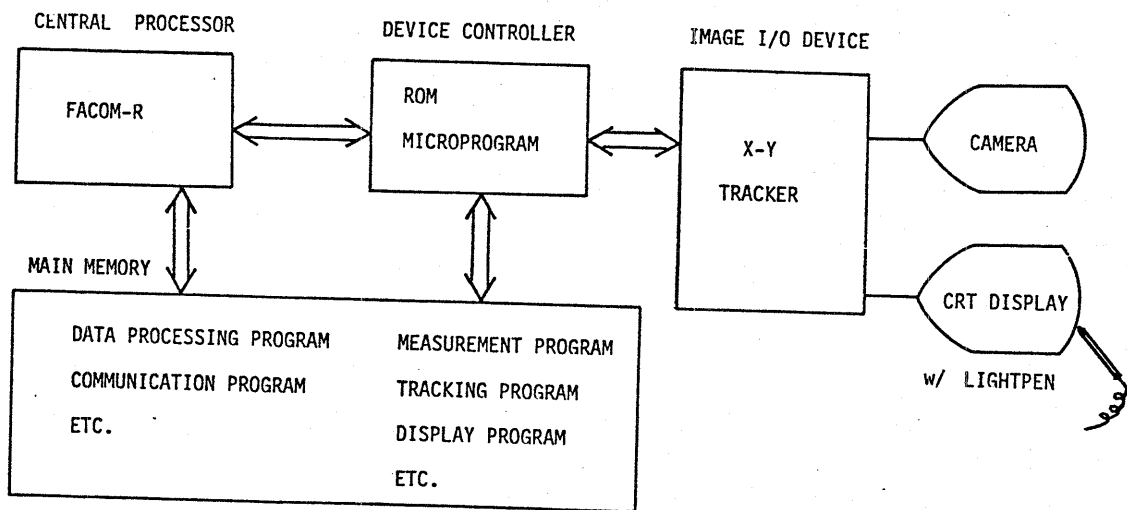


FIG 2.1 GENERAL PURPOSE IMAGE INPUT/OUTPUT SYSTEM

2.1 カメラ部

輝度を電気信号に変換し、外部から指定された条件にしたがってAD変換する。

イメージ・ディセクタの特徴は；

- 1 入出力特性の直線性がよい。
- 2 ダイナミック・レンジが広い。
- 3 速応性がある。
- 4 アパーチャ形状が任意に選べる。
- 5 出力は走査速度や履歴に無関係である。
- 6 感度を制限するのは量子化雑音のみである。
- 7 分解能が高い。

であり、乱走査可能で高速移動物体の撮像が可能である。

本装置では 1.5" 静電集束、静電偏向のイメージ・ディセクタ HTV-R312 を使用した。使用光電面は 15×15 mm²、アパーチャ径は 30 μmφ である。

なお、カメラ部および CRT の偏向入力は CPU およびトラッカ部より与えられプログラム制御によって任意に輝度測定や表示を行なうことができる。

2.2 CRT部

カメラまたは CPU より の信号を表示する。

CRT としては YHP 1301A を使用した。またライトペンを付加している。

2.3 トラッカ部

輝点を自動的に追跡する。追跡は、画面のモニタリングを目的としたラスタ・スキャンの帰線時間を使用して行ない、15.75 KHz の割合で X, Y 各 10 ビットの座標を出力する。追跡の開始座標は次のいずれかによって指定される。

- 1 全視野を走査し、最高輝度の点を見出し、その位置から追跡する。
- 2 CPU により与えられた初期値から追跡する。
- 3 ライトペンで指定された位置から追跡する。

もし、X-Y トラッカが追跡中の輝点を見失った場合は 1 の方法で輝点を求めて追跡を続行する。

なお、カメラ部、CRT 部およびトラッカ部を併せて映像入出力部と呼ぶことにする。この部分のブロック図を図 2.2 に示す。

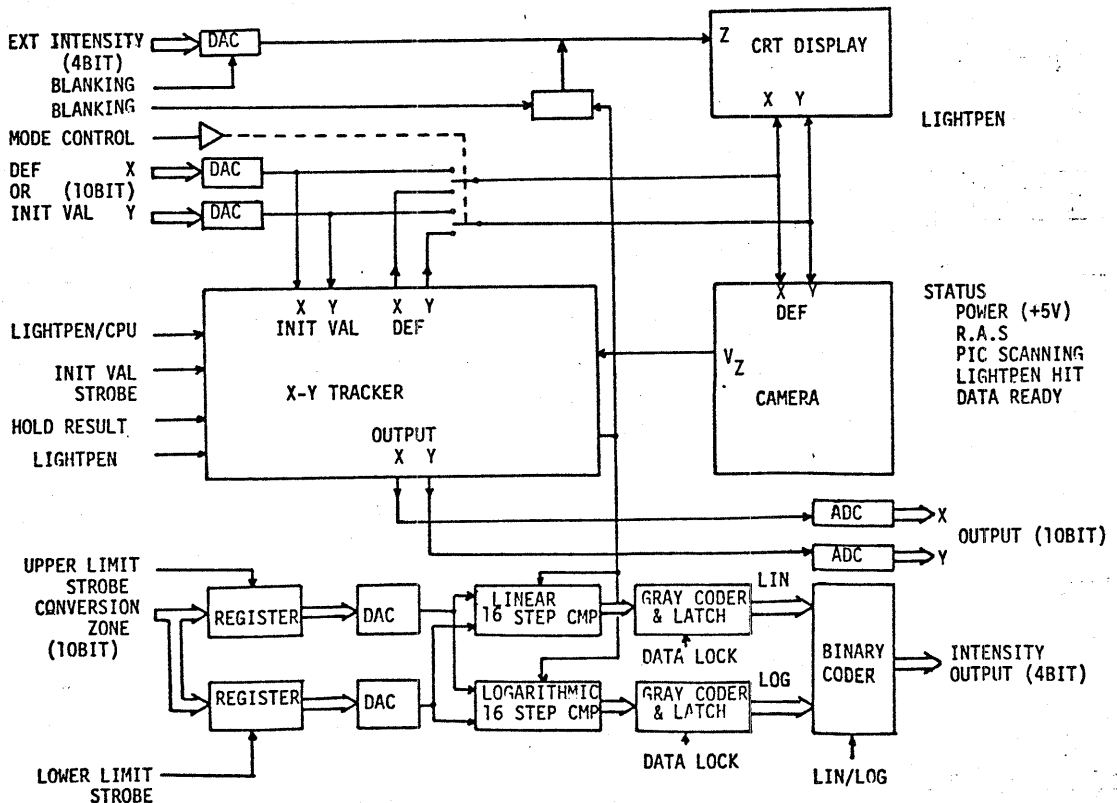


FIG 2.2 BLOCK DIAGRAM OF IMAGE INPUT/OUTPUT DEVICE

2.4 入出力制御部

カメラ部よりの入力とCRT部への出力を制御し、CPUとの通信を行なう。次章にて詳述する。

3 端末処理装置

ビデオ信号の入出力を伴うマン・マシン・インターフェースの制御は高速の処理を必要とするので、CPUが直接これを制御することは無理である。このため本装置の制御部がCPUの主記憶中に置かれた制御プログラムを自律的に解釈して入出力を行なう方式をとった。

この意味から、本制御部は独立の処理装置とみなすことができるので、端末処理装置と呼ぶことにする。

端末処理装置の機能としては次に示したものが必要とされる。これらの機能に対して、制御プログラムを構成するための命令が用意されている。

- 1 動作モードの設定
- 2 電子ビームの移動
- 3 制御の移動
- 4 CPUとの通信

命令の設計に関しては次の点に考慮を払っている。

- 1 プログラムの論理を格納番地と独立にし、リロケータブルにする。
- 2 手続きとデータとを分離する。
- 3 カメラ入力とディスプレイ出力のデータ形式の互換性を保つ。

このために、リロケーション・レジスタ、スタック・ポインタ・レジスタ、およびデータ・ディスクリプタを用いている。制御プログラムの構造を図3.1に示す。

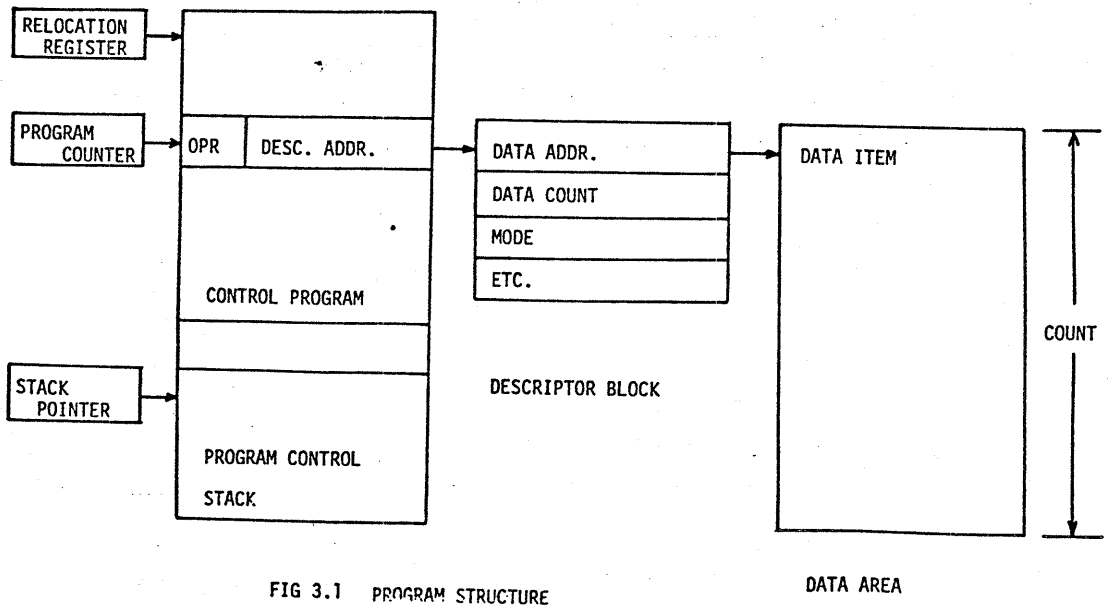


FIG 3.1 PROGRAM STRUCTURE

DATA AREA

3.1 端末処理装置の命令

本装置は前述のごとく、主記憶装置に格納された制御プログラムを自律的に解釈実行する。本節では制御プログラムを構成するための命令体系について概説する。付録1には命令、データ、およびディスクリプタの詳細を示しておく。

1 No-Operation (NOP)

何もせず単に次のステップに進む。

2 動作モードの設定

2.1 ディスプレイ・モードの設定 (Display Mode Set : DMS)

指定された輝度で点、ベクトル(および文字)の表示を行なうように条件を設定する。実際のディスプレイはこのモードの設定後に実行されるビーム移動命令により行なわれる。なお、カメラからの入力信号のモニタリングは停止される。

2.2 トラッキング・モードの設定 (Tracking Mode Set : TMS)

画面上の輝点をCPUまたはライトペンで指定された位置から開始して追跡し、そのX、Y座標をサンプリングした後主記憶に転送する。計測速度は最高 15.75 k点/sである。

2.3 輝度測定モードの設定 (Intensity Measurement Mode Set : IMS)

指定された条件で輝度測定を行なうように設定する。実際の輝度測定はこのモードの設定後に実行されるビーム移動命令により行なわれる。測定条件にはダイナミック・レンジの指定、AD変換の関数などがある。測定の終了は次の条件の検出による。

- i 指定されたデータ数の終了
- ii 測定範囲の走査終了
- iii 指定輝度以上の輝点の発見

輝点を発見すると端末処理装置はCPUに割り込みをかけ、実行を停止する。

3 ビームの移動

ビームの移動を行なう。ベクトルをディスプレイするときにはDDA方式によって直線を生ずる。

3.1 絶対座標によるビーム移動 (Move Beam Absolute : MBA)

3.2 相対座標によるビーム移動 (Move Beam Relative : MBR)

3.3 ラスタ・スキャンによるビームの走査

X軸またはY軸方向に指定された密度で指定された大きさの正方領域のラスタ・スキャンを行なう。

(3.4 文字の表示 (Move Beam Character : MBC))

今回の計画では実装していない。

4 制御の移動

4.1 無条件ジャンプ (JUMP)

4.2 条件判定 (Test Condition : TEST)

指定した条件が成立していればプログラム・シーケンスは次のステップに進み、成立していなければステップは次の次のステップに進む。条件には演算結果および入出力装置の状態を含めることができる。

4.3 プッシュ・ジャンプ (PUSH) サブルーチンの呼出しを行なう。

4.4 ポップ・ジャンプ (POP)

サブルーチンからの復帰を行なう。

5 CPUとの通信 (Send Interrupt : INT)

CPUに対して割り込みをかける。本命令を実行すると端末処理装置は実行を停止する。

3.2 端末処理装置の起動と停止

本装置はCPUによって起動される。起動された後は前述した命令を組合せて構成された制御プログラムをCPUの動作とは独立に実行し、割り込み命令 (Send Interrupt) の実行または輝点発見による割り込みを行なって停止する。

CPUによる端末処理装置の起動は入出力起動命令 (SIO) の実行によって行なわれる。端末処理装置の起動にはプログラム・カウンタ PC やリレジスタ REL、スタック・ポインタ SP の指定をあらかじめ行なっておく必要がある。これらの準備のためおよび、割り込みの発生によって端末処理装置の状態を調べるために CPU の入出力命令が使用される。

われわれは CPU として FACOM-R を使用しているが、FACOM-R の端末処理装置に対する入出力命令の機能は次に述べるとおりである。命令の詳細は付録 2 に示す。

1 起動および起動準備のための命令

1.1 SIO

CPU の A レジスタに用意された PC の値を端末処理装置に転送して、装置を起動する。

1.2 WRA

CPU の A レジスタに用意された REL の値を端末処理装置に転送する。

1.3 CTL

CPU の A レジスタに用意された SP の値を端末処理装置に転送する。

2 状態調査のための命令

2.1 AKI

割り込みを要求した制御装置の装置機番が CPU の A レジスタに設定される。

2.2 SNS

端末処理装置の装置状態語 1 DSW1 を CPU の A レジスタに読む。
DSW1 には割り込み発生時の装置の状態情報と X 軸偏向座標とが格納されている。

2.3 SNI

端末処理装置の装置状態語 2 DSW2 を CPU の A レジスタに読む。
DSW2 には割り込み発生時の装置の状態情報と Y 軸偏向座標とが格納されている。

2.4 RDA

端末処理装置の PC の値を CPU の A レジスタに読む。

3.3 端末処理装置の構成

映像入出力の制御プログラムを自律的に解釈実行するためには簡単な処理能力が必要であるが、本装置のように制御方式そのものが研究の対象となっている場合には命令の機能そのものにもある程度の可変性・柔軟性が要求される。また処理の対象がビデオ信号であり、マン・マン・インタフェースとして実時間の応

答が要求される。これらの条件を満たすために端末処理装置の制御はフォーウェア方式とした。

端末処理装置は図3.2のように構成した。

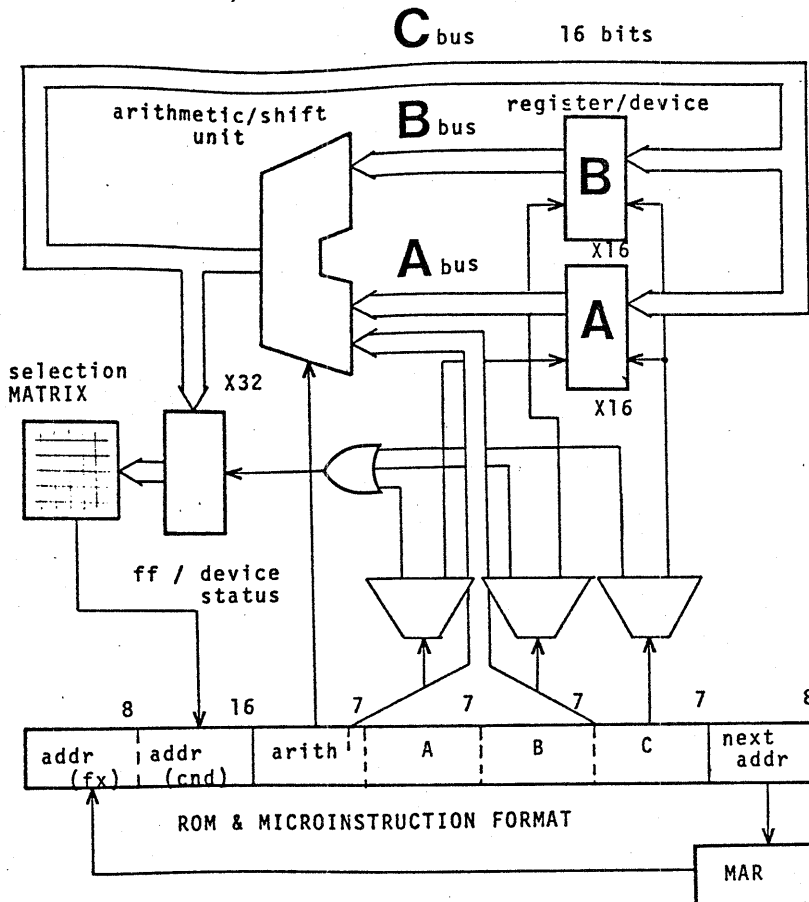


FIG 3.2 STRUCTURE OF DEVICE CONTROLLER

3.3.1 レジスタ

本処理装置の処理は装置の内外を問わずレジスタおよびフリップ・フロップとみなして演算および制御が行なわれる。汎用レジスタは16ビットのレジスタを最大32個有し、16個ずつA群レジスタとB群レジスタに分けられる。命令の解釈・実行、カメラ/CRTのデータ等の処理はこれらのレジスタを介して行なわれる。これに加えて6ビットのレジスタ3個が用意され演算結果によってプログラムを制御するために使用される。

3.3.2 フリップ・フロップ

端末処理装置の状態や外部装置の制御のためにフリップ・フロップを用意している。これらのフリップ・フロップには次りもりがある。

i) プログラム制御フリップ・フロップ

32個のJ-Kフリップ・フロップで構成され、命令で任意にセット、リセ

ット、反転を行なうことができる。1つの命令では最大3個のフリップ・フロップのJまたはK入力を指定することができる。これらのフリップ・フロップの出力は;

- a CPUとの通信を制御するため、
- b 映像入出力部を制御するため、
- c プログラムの制御のため、

に用いられる。

i) 状態制御フリップ・フロップ

16個のフリップ・フロップより成る。このフリップ・フロップは命令によって直接セット、リセットされるのではなく、演算結果、CPUの状態、映像入出力部の状態により値が定まるものである。このフリップ・フロップはプログラムの制御のために使用される。

3.3.3 マイクロ命令の機能と制御記憶

本処理装置に要求される処理速度は1命令当り100~200msである。しかも条件判定や制御の移動などで余分の処理時間を費したくないので;

- 1 2項演算のソースとデスティネーションを一度に指定すること、
 - 2 分岐先を命令の中に含めること、
- とし、命令を4番地形式とした。さらに;
- 1 多分岐の条件分岐を容易にすること、
 - 2 条件判定とそれに伴う分岐のために余分の時間と余分のプログラム・ステップを必要としないこと、
 - 3 任意個数の多分岐を可能とすること、
 - 4 条件符合せのために余分のプログラム・ステップを必要としないこと、
 - 5 マイクロ・サブルーチンの組込みを容易とすること、
 - 6 プログラムのデバッグと修正を容易にするための命令の格納を物理的な場所と無関係とすること、
 - 7 制御記憶の必要容量を小さくすること、

の条件から命令の番地指定を命令の次番地部の指定に条件部を加えて“連想記憶”方式とした。また、命令はプログラムを容易とし記憶容量を小さくするためにエンコード形式とした。

命令の形式は図3.2に示す。

マイクロ命令の種類は付録3に示す。

本装置のファームウェアはダイナミック方式とすることも検討したが、番地指定方式の特徴を生かした上で性能的ならびに経済的に実現することに難があったこと、高速で安価なRAMが入手困難であったことから行なわれないこととした。

§3.1~3.2の機能を組込むために要したマイクロ命令のステップ数は約210ステップであり、きわめて小容量であることに注目してほしい。付録4にコーディングの一部を示しておく。

制御記憶のための記憶素子としては

- 1 番地指定の条件部が存在しないものはフルデコード形式のROMを用い、
- 2 条件部が存在するために“連想記憶”とした部分はデコード部とメモリ部共にNANDゲートを用いた。この場合記憶内容はゲート入力の接続のあり/なしに対応させている。図3.3参照

この結果素子としては、アクセス時間は NAND 2 段分の遅延時間であり、超高速の処理を可能にしている。われわれの場合 バスの布線や終端に特に神経をいかなかったし、パイプライン制御等の技法も用いていないのでアクセス時間は約 40 ns、システムとしてのマッシュ・サイクル時間は 150 ns となっている。

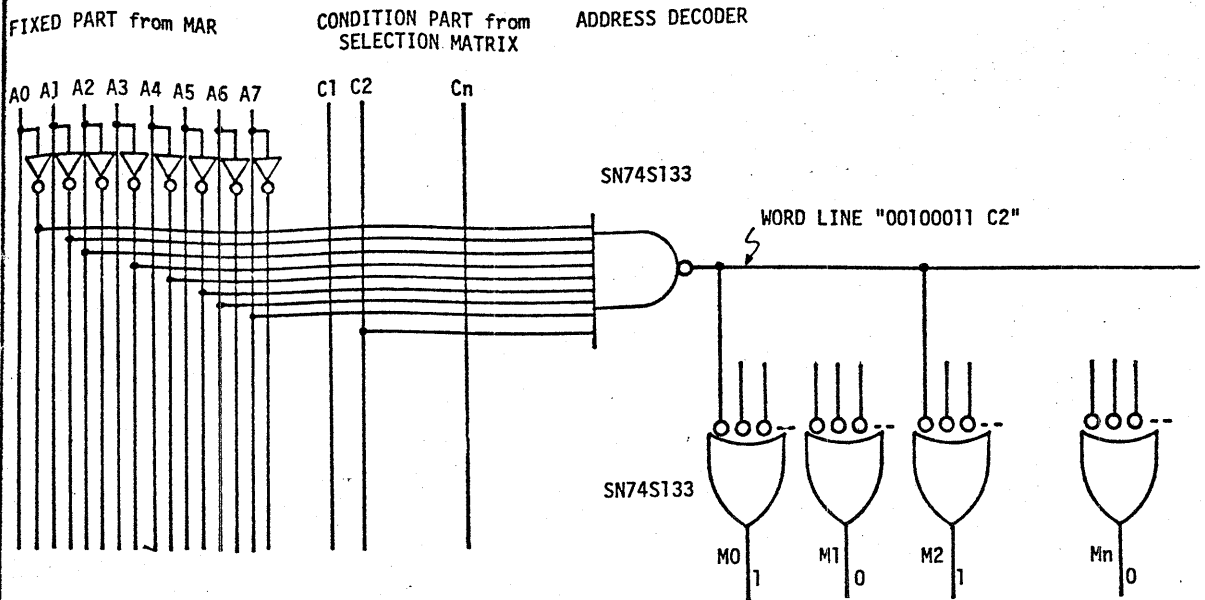


FIG 3.3 ROM ORGANIZED BY NAND GATES

4 おわりに

ビデオ入出力を伴うマン・マシン・インターフェースの構成とその制御プログラムによる制御方式およびファームウェア方式による制御装置の構成について述べた。

本汎用映像入出力装置の制御プログラムの作成は各利用者に任されるが、広範囲の応用に使用できるものと考えている。

4 番地形式のマイクロ命令と連想方式によるアドレス方式はマイクロ・ステップの流れを追跡するという点では多少の困難があり、ファームウェアのデバッグの面からは問題を残してはいるが、本装置のような高速のマイクロ・プログラム実行の組込みに適している。

本研究は文部省科学研究補助金の交付を受けて行なっているものである。

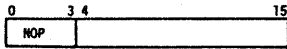
本装置の演算部の初期の設計は本学学生岡田信君が行なった。イメージ・ディスプレイおよびトラッカ部の作成は浜松テレビの倉沢、伊井の両氏が受持たれた。各位の御甚かに対して謝意を表す次第である。

参考文献

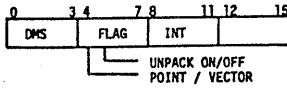
- (1) 鈴木倉沢：イメージ・ディスプレイとその計測用テレビジョンへの応用，テレビジョン vol. 25, No 7, pp. 558~568 (1971).
- (2) ITT Technical Note 112, "A Survey of Image Dissector Performance Characteristics" (1970).
- (3) Horn, B.K.P: The Image Dissector "eyes" MIT A.I. Memo, No 178, Aug., 1969.
- (4) Desmond, W.H: A Conversational Graphic Data Processing System (Prentice-Hall, 1969).
- (5) Newman, W.M., Sproul, R.F.: Principles of Interactive Computer Graphics (McGraw-Hill, 1972).

付録1 映像入出力装置の命令とデータの形式

NO-OPERATION

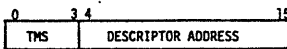


DISPLAY MODE SET



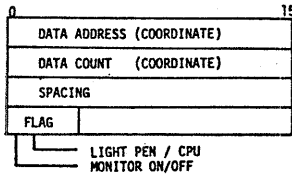
- カメラ信号をoffとする。
- unpackがoffの時INTで定められた輝度で表示する。
- 実際のディスプレイは以後に続くビームの移動で行なう。

TRACKING MODE SET



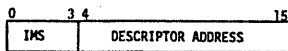
- トラッキングモードで輝度の追跡を行なう。

DESCRIPTOR ADDRESS



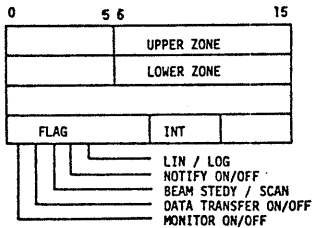
- 測定データは Spacing で指定された間隔でリアルタイムに data address から記憶に data count 語数だけ格納する。
- データの形式は DATA FORMAT A を使用する。
- 追跡開始の位置は、ライトペンで指示する。または本命令実行以前のビームの位置より開始する。

INTENSITY MEASUREMENT MODE SET



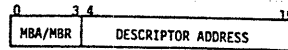
- 輝度測定を行なう。
- 実際の測定はモード設定後のビームの移動により開始する。

DESCRIPTOR ADDRESS



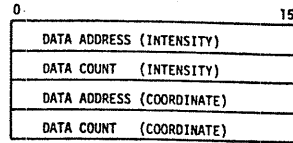
- 変換のためのゾーンを設定する。
- 測定の場合はFLAGに基づいて行なう。
- 測定輝度は DATA FORMAT C の形式で主記憶に転送される。
- notifyがonのときINTで指定された輝度と測定輝度の比較が行われ、指定輝度以上のものを発見したとき、測定みを停止して停止する。

MOVE BEAM ABSOLUTE / RELATIVE



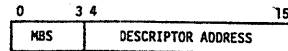
- ビームの移動を行なう。

DESCRIPTOR ADDRESS



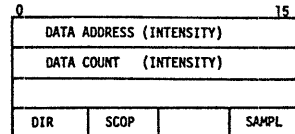
- 絶対座標指定の場合は DATA FORMAT A を使用する。
- 相対座標指定の場合は DATA FORMAT B を使用する。
- 輝度測定モード設定でデータ転送を行なう場合とディスプレイ設定でunpack flagがonの場合は DATA FORMAT C を使用する。

MOVE BEAM SCAN



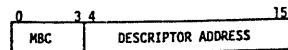
- ラスタスキャンを行なう。

DESCRIPTOR ADDRESS



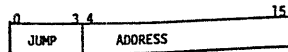
- DIRの指定によりX又はY方向にラスタスキャンを行なう。
- SCOPの指定により一辺の長さを定める。
 $2^i \quad i=1, 2, \dots, 10$
- SAMPLの指定により隣接画の間隔を定める。
 $2^i \text{ およ } i=1, 2, \dots, 6$
- 命令の終了は data count = 0 または指定範囲の Scan 終了時。

MOVE BEAM CHARACTER



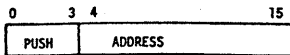
- 文字ストロークを行なう (今回は実装しない)

JUMP



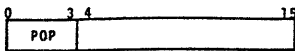
C (PROGRAM COUNTER) := EFF. ADDR.

PUSH JUMP



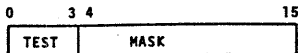
STACK POINTER := STACK POINTER + 1
 C(STACK POINTER) := C(PROGRAM COUNTER)
 C(PROGRAM COUNTER) := EFF.ADDR.

POP JUMP



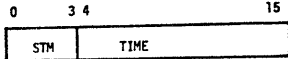
C(PROGRAM COUNTER) := C(STACK POINTER)
 STACK POINTER := STACK POINTER - 1

TEST CONDITION



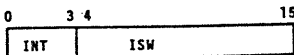
- MASKが"1"である条件が成立すればプログラム・シーテンスは次のステップへ進み"0"であれば1命令スキップされて、次のステップへ進む。
- MASKの内容は、カウンタの状態出力(ライト・イベント、データ、パワー、等)、演算結果の零、正、負、等。

START TIMER



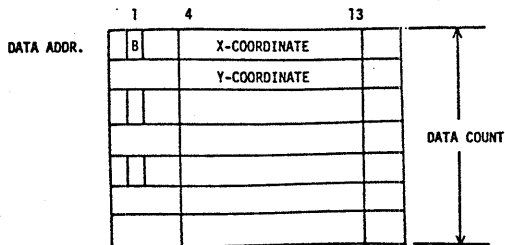
- 次の期計画には含まない。

SEND INTERRUPT

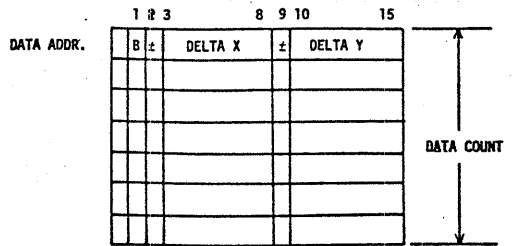


- プログラム制御の割出しと主処理装置に行ない動作を停止する。

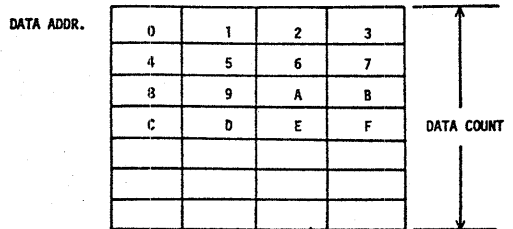
DATA FORMAT A



DATA FORMAT B



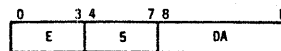
DATA FORMAT C



付録2 映像入出力装置制御のための命令

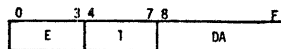
1. 映像入出力装置制御・起動命令

1.1 装置の起動：SIO命令



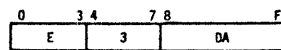
Aレジスタにあらかじめ用意されたプログラム・カウンタPC(映像入出力装置が使用する)の値を映像入出力装置に転送して、同装置を起動する。
 FACOM-Rのプログラム・シーテンスは次の次のステップに進む。

1.2 リローション・レジスタの設定：WRA命令



Aレジスタにあらかじめ用意されたリローション・レジスタRELの値を映像入出力装置へ転送する。
 プログラム・シーテンスは次の次のステップへ進む。

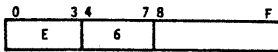
1.3 スタック・ポインタの設定：CTL命令



Aレジスタにあらかじめ用意されたスタック・ポインタSPの値を装置へ転送する。
 プログラム・シーテンスは次の次のステップへ進む。

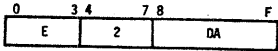
2. 映像入出力装置の状態の検査の命令

2.1 割込み検査の調査: AK1命令



Aレジスタへ割込み要求した装置検査の結果を送る。

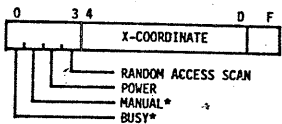
2.2 状態の検査 1: SNS命令



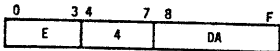
a. 映像入出力装置の装置状態情報 DSW1 の Aレジスタへ
 転送される。

プログラム・シーケンスは次の次のステップへ進む。

b. DSW1 FORMAT



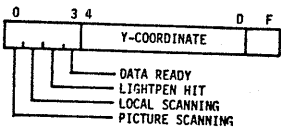
2.3 状態の検査 2: SN1命令



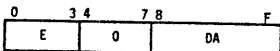
a. 映像入出力装置の装置状態情報 DSW2 の Aレジスタへ
 転送される。

プログラム・シーケンスは次の次のステップへ進む。

b. DSW2 FORMAT



2.4 PCの検査: RDA命令



映像入出力装置のプログラムカウンタ PCの値を Aレジスタへ
 転送する。

プログラム・シーケンスは次の次のステップへ進む。

付録3 マイクロ命令一覧表

OPERATION PART

SELECT	FUNCTION	COMMENT
B1 B2 B3	B4 B5 B6 B7	
0 0 0	see below	ARITHMETIC MODE WITH CARRY
0 0 1	see below	ARITHMETIC MODE WITHOUT CARRY
0 1 0	see below	LOGIC MODE
0 1 1	C0 C1 C2 C3	IMMEDIATE CONSTANT OF 16 BITS
1 0 0	* * * *	1-BIT LEFT SHIFT CIRCULAR
1 0 1	* * * *	2-BIT LEFT SHIFT CIRCULAR
1 1 0	* * * *	4-BIT LEFT SHIFT CIRCULAR
1 1 1	* * * *	8-BIT LEFT SHIFT CIRCULAR
		LOGIC MODE
		ARITHMETIC MODE
		WITHOUT CARRY
		WITH CARRY
see above	0 0 0 0	A
	0 0 0 1	A+B
	0 0 1 0	A+B
	0 0 1 1	0
	0 1 0 0	A.B
	0 1 0 1	(A+B).PLUS.(A.B)
	0 1 1 0	A.MINUS.B
	0 1 1 1	(A.B).MINUS.1
	1 0 0 0	A.PLUS.(A.B)
	1 0 0 1	A.PLUS.B
	1 0 1 0	(A+B).PLUS.(A.B)
	1 0 1 1	(A.B).MINUS.1
	1 1 0 0	A.PLUS.A
	1 1 0 1	(A+B).PLUS.A
	1 1 1 0	(A+B).PLUS.A
	1 1 1 1	A
		A.PLUS.1
		(A+B).PLUS.1
		(A+B).PLUS.1
		ZEPO
		A.PLUS.(A.B).PLUS.1
		(A+B).PLUS.(A.B).PLUS.1
		A.MINUS.B
		A.B
		A.PLUS.(A.B).PLUS.1
		A.PLUS.B.PLUS.1
		(A+B).PLUS.(A.B).PLUS.1
		A.B
		A.PLUS.A.PLUS.1
		(A+B).PLUS.A.PLUS.1
		(A+B).PLUS.A.PLUS.1
		A

A,B,C PART

B1	B2	B3	B4	B5	B6	B7	COMMENT
0	0	0					REGISTER NO. reserved
0	0	1					REGISTER NO. A-GROUP REGISTER
0	1	0					REGISTER NO. B-GROUP REGISTER
0	1	1					REGISTER NO. reserved
1					0		FLIP-FLOP NO. 0 FLIP-FLOP K-INPUT (RESET)
1					1		FLIP-FLOP NO. 1 FLIP-FLOP J-INPUT (SET)

付録4 マイクロ・ルーチンの例

/* INSTRUCTION FETCH ROUTINE */

no.	fixed part	conditional part	operation	next address
031	FETCH		MAR := PC + REL	
032	FETCH+1		PC := PC + 1	
033			reset DRCT, FO	
034			reset F1, F2, F3	DMA
035	RTRN	SRVO,FO,FT,FZ,F3	IR := HDR, set FO	EXEC

/* DIRECT MEMORY ACCESS */

036	DMA-1		MAR := MAR + 1	DMA
037	DMA	SRVO	CPU := MAR, set ADRI	DMA
038	DMA	SRVO	reset ADRI	W.SRVO2
039	W.SRVO2	SRVO		D.SRVI
040	D.SRVI	SRVO, DRCT	CPU := HDR, set SRVI	D.SRVI

CAH