

---

**ADVANTEST®**

株式会社 アドバンテスト

---

**取扱説明書**

**TR4720-570**

R6502用 パーソナリティ・キット

MANUAL NUMBER 0059 A 104

---

禁無断複製転載

© 1981 株式会社アドバンテスト

---



# 目 次

第1章 概 説		
1-1	概 要 .....	1-1
1-2	パーソナリティ・キットの構成 .....	1-2
1-3	性能諸元 .....	1-3
第2章 セット・アップ		
2-1	パーソナリティ・キットの交換方法 .....	2-1
2-1-1	パーソナリティ・ボードの交換 .....	2-1
2-1-2	パーソナリティ・パネルの交換 .....	2-2
2-1-3	CPU プローブの交換 .....	2-2
2-2	$\mu$ P への接続方法 .....	2-3
第3章 パーソナリティ・パネルの使用方法 .....		3-1
3-1	<b>TRIGGER QUALIFIER</b> スイッチの使用方法 .....	3-1
3-2	<b>TRACE THEN HALT</b> スイッチの使用方法 .....	3-3
3-3	CPU ステイタス LED の使用方法 .....	3-4
3-4	<b>TRACE THEN HALT</b> 使用時の表示上の注意 .....	3-4
第4章 データの取込みおよび表示に関する情報 .....		4-1
4-1	データの取込みタイミング .....	4-1
4-2	取込まれたデータの表示に対するコメント .....	4-10
4-2-1	逆アセンブル・フォーマットについて .....	4-10
4-2-2	$\mu$ P の特別な動作における表示について .....	4-11
第5章 動作チェック .....		5-1



# 第 1 章 概 説

## 1-1 概 要

**TR4720-570** は、**TR 4720** ロジック・アナライザと組合わせて、**R6502**  $\mu$ P ( マイクロプロセッサ ) を使用した各種システムの総合的なデバッグを行なうことができます。このパーソナリティ・キットの特長は、以下の通りです。

1. 各種  $\mu$ P に対して専用のパーソナリティ・キットが用意されていますので、測定  
のセット・アップが容易で、データの取込みが迅速に行なえます。
2. 逆アセンブル機能によって  $\mu$ P 固有のニーモニックで表示できますので、得られ  
たデータの解析が非常に容易となります。
3. データ・バス・クォリファイアにより本体のトリガ機能が、さらに強化されます。
4.  $\mu$ P の信号に疑似したテスト・パターン発生器を内蔵していますので、CPU プ  
ローブのテストから、本体の基本測定機能のテストまで行なえます。

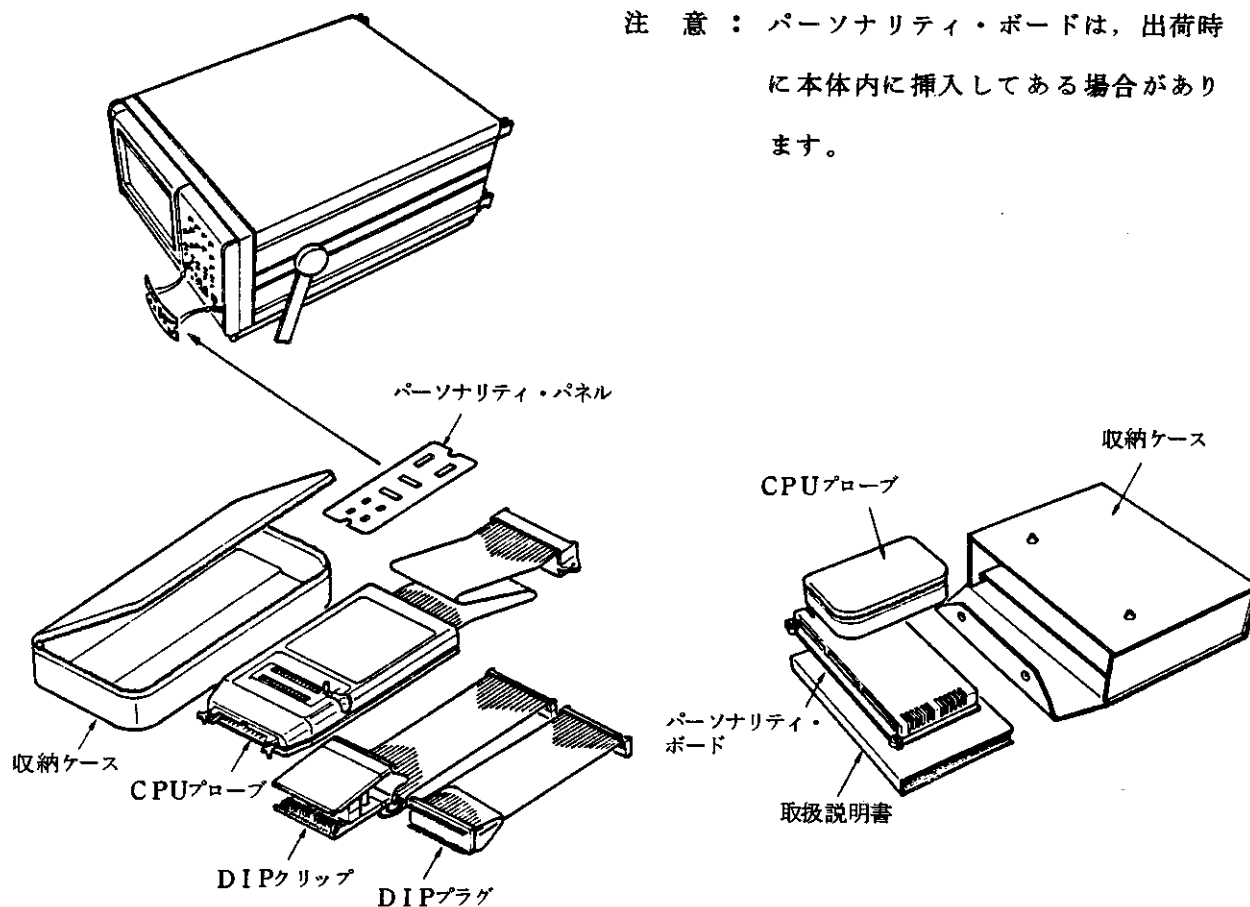
### NOTE

本取扱説明書は、対象となる  $\mu$ P の種類に依存する部分を中心に説明してあり  
ます。  $\mu$ P の種類に依存しない基本的な機能、操作方法などは **TR4720** 本  
体取扱説明書をご覧ください。

1-2 パーソナリティ・キットの構成

パーソナリティ・ボード(本体内に挿入)	1
CPUプローブ(被測定システムと本体とを接続)	1
プローブ用コネクタ	
40ピンDIPクリップ・コネクタ	1
40ピンDIPプラグ・コネクタ	1
パーソナリティ・パネル(本体の正面パネルに装着)	1
CPUプローブ収納ケース	1
パーソナリティ・キット収納ケース	1
取扱説明書	1

注 意 : パーソナリティ・ボードは、出荷時に本体内に挿入してある場合があります。



1-3 性能諸元

適 合  $\mu$ P : ROCKWELL R6502 および相当品

クロック周波数 : 被測定システムによる。max. 2MHz

入 力 電 流 :  $-200\mu\text{A}$  max. (ロー・レベル)

$20\mu\text{A}$  max. (ハイ・レベル)

クォリファイア : ① OP または OFF

② READ, WRITE または OFF

(①と②はORされる)

CPU ステータス表示 :

HALT ;  $\mu$ P の READY がローのとき点灯

HALT BY LA ; TR 4720 より  $\mu$ P に HALT 信号を出した

ときに点灯

TRACE THEN HALT :  $\mu$ P をプローブに装着してトレース・モードのとき動作

し, トレース終了後  $\mu$ P の READY ラインをローにする。





## 第2章 セット・アップ

### 2-1 パーソナリティ・キットの交換方法

既に装着されているパーソナリティ・キットと、測定対象となる $\mu$ Pが異なる場合には、対象となる $\mu$ P用のパーソナリティ・キットとの交換が必要です。交換が必要なのは、パーソナリティ・ボード、パーソナリティ・パネルとCPUプローブの3点です。以下の手順で交換して下さい。

#### 2-1-1 パーソナリティ・ボードの交換

1. **POWER** スイッチを**OFF** に設定します。
2. 本体上カバーの4本のネジ(3%, +)を外し、上カバーを取除きます。
3. 正面パネルから見て右端のボードがパーソナリティ・ボードです。パーソナリティ・ボード上面の3個のコネクタを外します。
4. パーソナリティ・ボード両側のボード・ハンドルを用いて、ボードを取外します。  
〔図-1参照〕
5. 交換するパーソナリティ・ボード〔写真1参照〕の銘板が外側になるようにして、スロット両側のボード・ガイドに従って挿入して下さい。
6. パーソナリティ・ボード上面の3個のコネクタを取付けます。
7. 本体上カバーを4本のネジで取付けます。
8. **POWER** スイッチを**ON** に設定しますと、交換したパーソナリティ・ボードの対象 $\mu$ P名が、約30秒後にCRTディスプレイに表示されますので確認して下さい。〔写真3参照〕
9. 取外したパーソナリティ・ボードは、所定のパーソナリティ・キット収納ケースに入れて下さい。

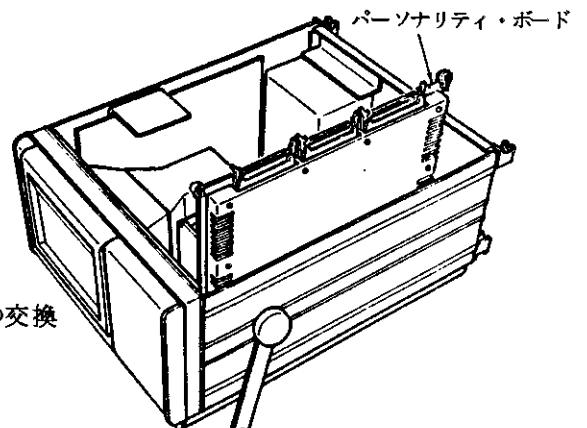


図-1 パーソナリティ・ボードの交換

### 2-1-2 パーソナリティ・パネルの交換

1. 正面パネル右下側に取付けられているパネルが、パーソナリティ・パネルです。  
まず **TRIGGER QUALIFIER** スイッチの設定を、**READ**と**MEM.**にします。  
次に CPU ステータス LED 側から、パーソナリティ・パネルをたわめると、パネルはプラスチックで作られていますので、取外すことができます。
2. 交換するパーソナリティ・パネルの切り欠き部分を飾りネジに挿入することで、取付けます。(図-3 参照) 取付けた後に、**TRIGGER QUALIFIER** スイッチを全て **OFF** に設定します。
3. 取外したパーソナリティ・パネルは所定の CPU プロブ収納ケースに入れて下さい。

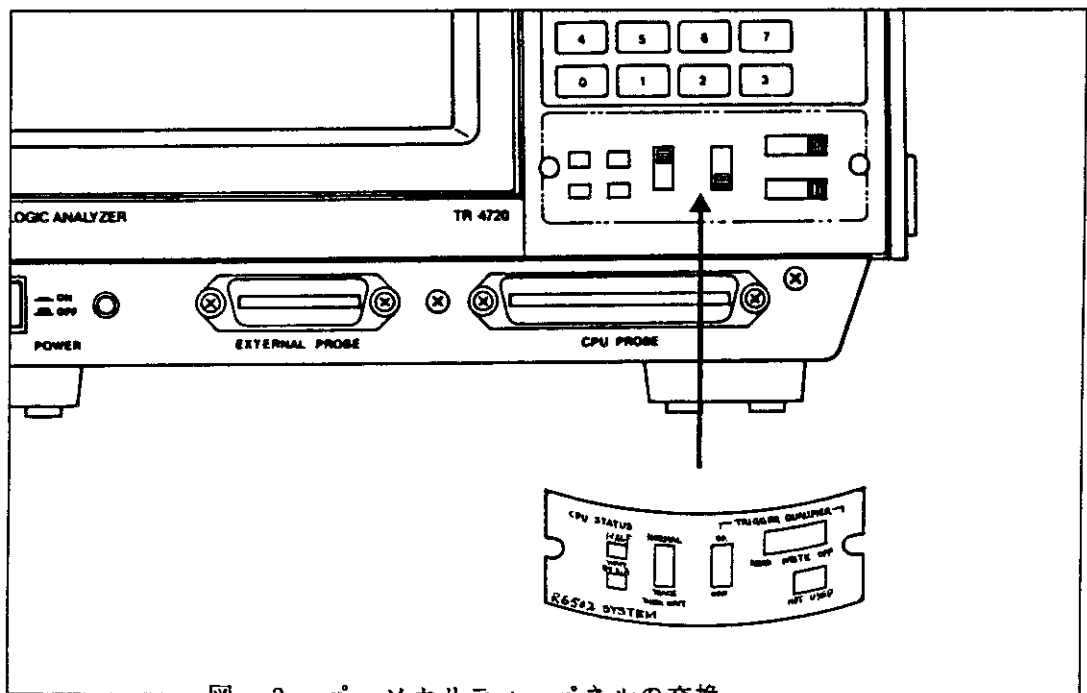


図-3 パーソナリティ・パネルの交換

### 2-1-3 CPU プロブの交換

1. 交換したパーソナリティ・ボードに対応した CPU プロブ (写真 2 参照) を正面パネル下面の CPU プロブ・コネクタへ取付けて下さい。
2. 取外した CPU プロブは所定の CPU プロブ収納ケースに入れて下さい。

## 2-2 $\mu$ P への接続方法

CPU プロブを SUT ( System Under Test ) の  $\mu$ P へ接続するために 2 種類のコネクタが用意されています。接続の様子を [ 図-4 ] に示します。

- 1) 40 ピン DIP クリップ・コネクタの場合…………… SUT の  $\mu$ P チップをはさみ込んで使用します。この場合、本器から  $\mu$ P の一時的な動作停止 (**TRACE THEN HALT** 機能) を指示することはできません。
- 2) 40 ピン DIP プラグ・コネクタの場合…………… SUT の  $\mu$ P がソケットを使用している場合に使用できます。まず、 $\mu$ P をソケットから外し、「CPU PROBE」上のソケットに装着します。次にプラグ・コネクタを  $\mu$ P のソケットに挿入します。この場合 **TRACE THEN HALT** 機能 [ 3-2 項参照 ] を使用できます。

いずれの場合でも、ピンの向きに注意して下さい。

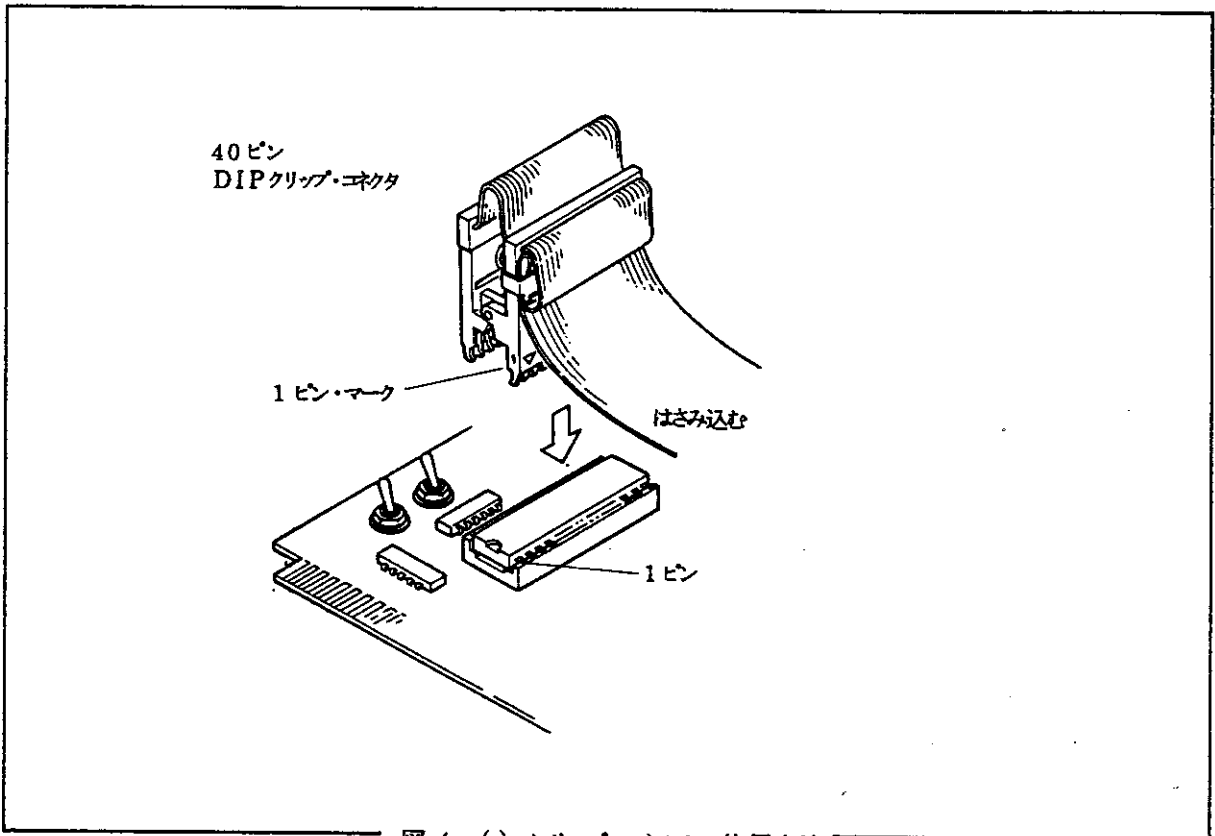


図 4-(a) クリップ・コネクタの使用法

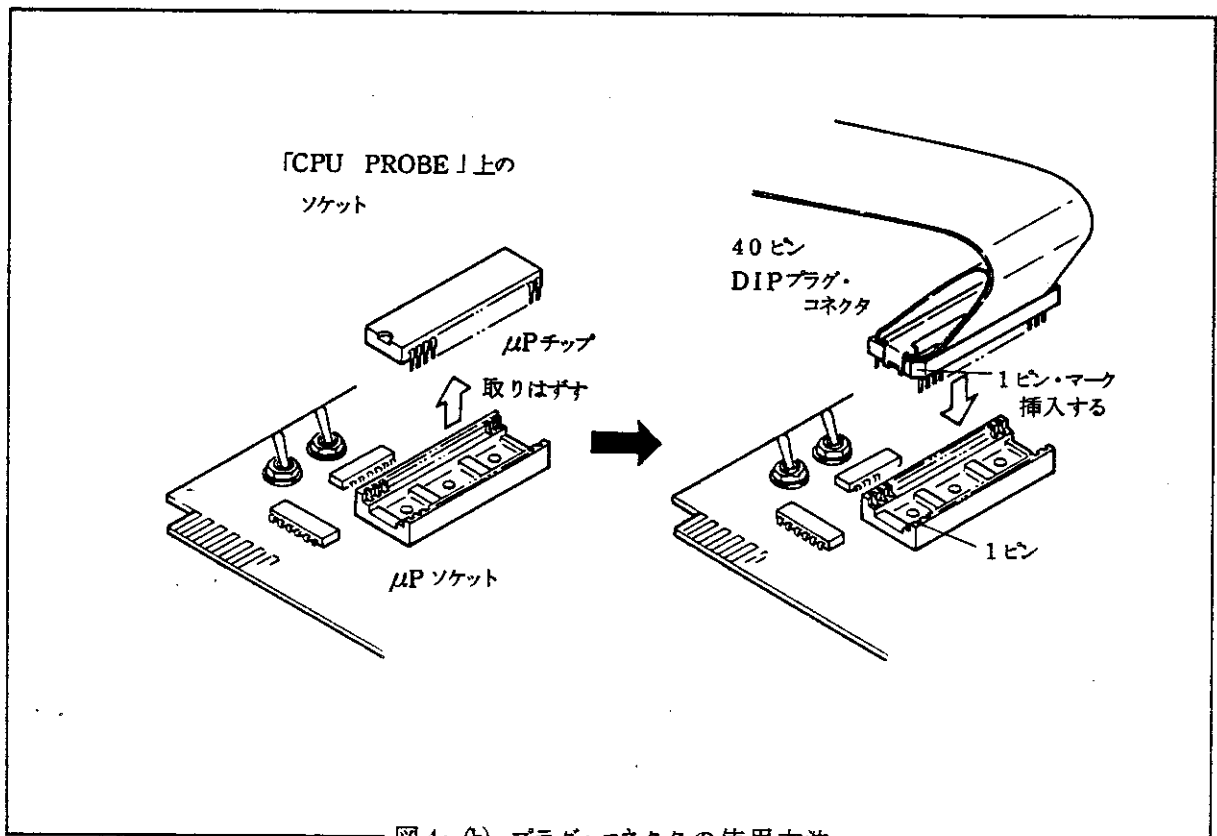


図 4-(b) プラグ・コネクタの使用法

### 第3章 パーソナリティ・パネルの使用法

パーソナリティ・パネルを〔図-5〕に示します。パネルは、**TRIGGER QUALIFIER** スイッチ ( $S_1, S_2, S_3$ ), **TRACE THEN HALT** スイッチ ( $S_4$ ), CPU ステータス LED の3つの部分に分けることができます。

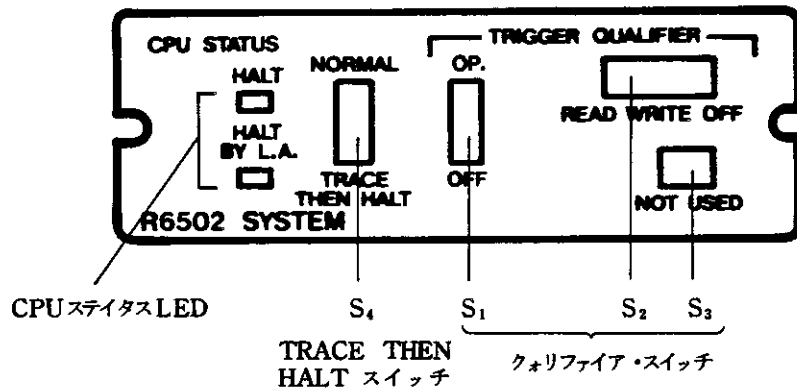


図-5 パーソナリティ・パネル

#### 3-1 TRIGGER QUALIFIER スイッチの使用法

CPU プローブからは、アドレス・バス、データ・バス上の各データと共に、コントロール信号も取込まれますが、パーソナリティ・ボード内でこれらの信号をデコードして、データ・バス・データの種別を示すデータ・バス・フラグが作られ、バス・データと共に「データ・メモリ」へ送られます。R6502 の場合、データ・バス・フラグは2 ch. で以下の通りです。

- フェッチ・サイクルのデータ (オペコード ; **OP**) かどうか。
- リード・データか、ライト・データか。

「ステート表示」の場合に、これらのフラグがパーソナリティ・ボード内のプログラムにより解読され CRT ディスプレイに表示されます。「アブソリュート表示」の例を〔写真4〕に示します。データ・バス・データの右側に表示されているのが、データ・バス・フラグの外部表現で以下の3種類あります。

- 1) /**OP** : SUTの  $\mu P$  がオペコードであると解釈したデータです。

2) /RD :メモリ (RAMおよびROM)から $\mu$ Pへリードされたデータです。

3) /WR :メモリ (RAM)へ $\mu$ Pからライトされたデータです。

データ・バス・データのこれらの違いを、トリガ機能として使うのがトリガ・クォリファイアです。S<sub>1</sub>、S<sub>2</sub>をすべてOFFと設定すれば(S<sub>3</sub>はOFFに固定)、キーボードから設定したトリガ条件で測定が行なわれますが、2つのスイッチのうち1つでもOFFでなければクォリファイアが働きます。2つのスイッチの組合わせで

1)~3)の種類を識別する機能をトリガ条件に加えることができます。S<sub>1</sub>とS<sub>2</sub>の設定は「OR」されるため、R6502の場合、スイッチによる組合わせは6通りとなります。組合わせを〔表1〕に示します。「TIME INTERVAL」以外のすべての測定モードで有効ですが、特に「TRACE TRIGS」では便利です。

全く「トリガ条件」を設定しなくても、リード・データだけや、あるいはライト・データだけを取込むこともS<sub>1</sub>をOFF、S<sub>2</sub>をREADあるいはWRITEと設定するだけで行なえます。

クォリファイア機能	S <sub>1</sub>	S <sub>2</sub>
/RD を識別	OFF	READ
/WR を識別	OFF	WRITE
なし	OFF	OFF
/OP + /RD を識別	OP	READ
/OP + /WR を識別	OP	WRITE
/OP を識別	OP	OFF

S<sub>3</sub>は常にOFF

表1 トリガ・クォリファイア機能

### 3-2 TRACE THEN HALT スイッチの使用方法

本器の通常的使用方法では、SUTの $\mu P$ への介入は行ないませんが、プログラムの流れを切れ目なく調べたい場合にこのスイッチを使用します。このスイッチを使用すると、特定のトリガ・ポイントを基点として、1~256ステートの任意のステート数をトレースした後に、 $\mu P$ へHALTをかけることができます。この結果、任意のステート数毎にプログラムの動きを分割しながら調べることができます。この様子を〔図-6〕に示します。ただし、ライト・サイクルでは、表示できる場合と、できない場合がありますので、注意して下さい。詳しくは〔3-4項〕を参照して下さい。

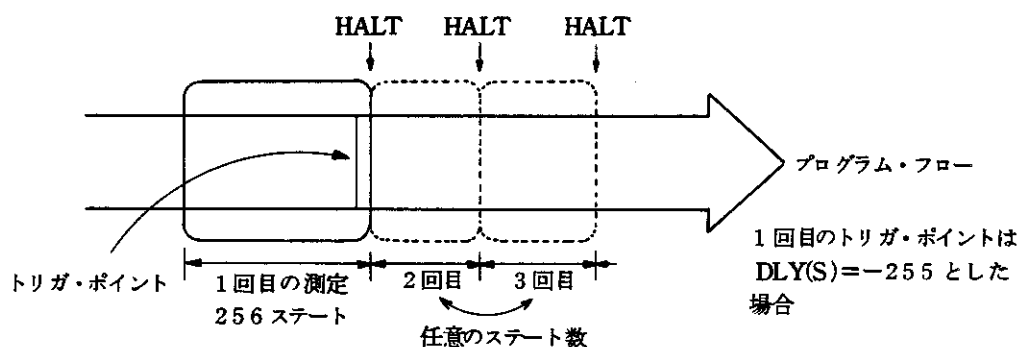


図-6 TRACE THEN HALT によるステート解析

以下の手順で使用して下さい。

1. 「TRACE THEN HALT」機能は40ピンDIPプラグ・コネクタを使用した場合にのみ使用できますので、SUTの $\mu P$ をソケットから外し、CPUプローブのソケットに挿入します。SUTの $\mu P$ 用ソケットへはプラグ・コネクタで接続します。〔2-2項参照〕
2.  $S_4$ をTRACE THEN HALTに設定し、キーボードから「トレース条件」を設定します。「DLY(S)」の設定は任意ですが、-255と設定した場合には、測定実行においてトリガ発生直後に $\mu P$ へHALTがかけられます。
3. RUNキーを押しますと、測定実行が開始され、トリガ発生によって実行が終了しますとHALTがかけられます。本器によるHALTはCPUステータスLEDのHALT BY L.A.の点灯によって確認できます。
4. 次に継続して実行したい場合は、TRACEキーを押して「トレース条件設定状態」とし、DEFAULTキーによって「トレース条件」をすべて「DON'T CARE」にします。次に「DLY(S)」の設定を行ないます。「DLY(S)」=0

の場合、**RUN** キーを押すことで256 ステートだけデータを取込みますと、**HALT** しますが、「**DLY(S)**」= $-n$  ( $n=1\sim 255$ )と設定しますと、**RUN** キーを押すことで $256-n$  ステートだけデータを取込んで、**HALT** がかけられます。 $n=255$  ですとシングル・ステート実行となります。

5. 「**TRACE THEN HALT**」機能を停止する時には、 $S_4$  を**NORMAL**に設定します。

### 3-3 CPU ステータス LED の使用方法

SUT の $\mu P$  はDMAやマルチ $\mu P$ などによって、プログラムの実行を中断している状態があります。このような状態ではトレースによる観測はできませんので、 $\mu P$  がどんな状態にあるかをLEDで表示しています。これがCPUステータスLEDで次の2種類があります。

- 1) **HALT** :  $\mu P$  のREADYラインがLowの場合に点灯します。
- 2) **HALT BY L. A.** : 本器から $\mu P$ へ**HALT**をかけている場合に点灯します。

### 3-4 TRACE THEN HALT 使用時の表示上の注意

1. ライト・サイクルの前にDiscarded Dataがあり、**READY**をこのDiscarded Dataサイクルでローに落とした場合は、次の実行のとき、ライト・データをサンプルし、表示することができます。〔図-7(A)参照〕
2. ライト・サイクル時に**READY**をローに落としますと、このサイクルは**HALT**できずに次のリード、またはオペコード・サイクルで**HALT**がかかります。したがって、ライト・サイクルは、サンプルも表示もできません。〔図-7(B)参照〕

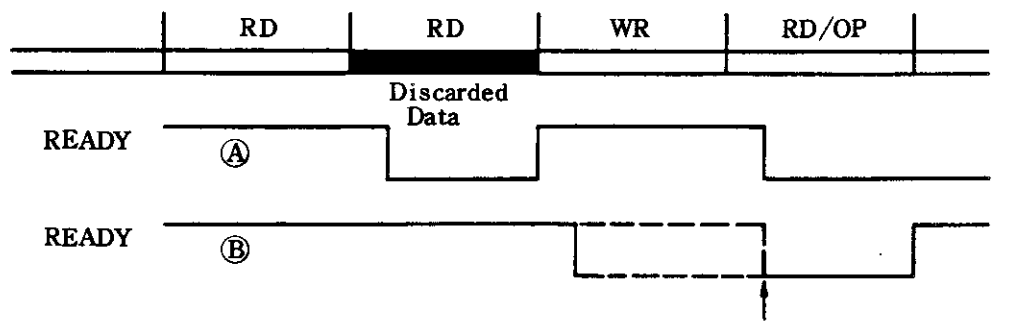


図-7 サンプルできるライト・サイクルと  
サンプルできないライト・サイクル



## 第 4 章 データの取込みおよび表示に関する情報

### 4-1 データの取込みタイミング

$\mu$ P からの信号のうち、パーソナリティ・ボードによってサンプルされるのは以下のデータです。

- o アドレス・バス・データ 16 ch.
- o データ・バス・データ 8 ch.
- o データ・バス・フラグ 2 ch.

F 1 = 1 (オペコード・フェッチ・サイクル)

= 0 (上記以外)

F 3 = 1 (R/W信号がハイ;リード・サイクル)

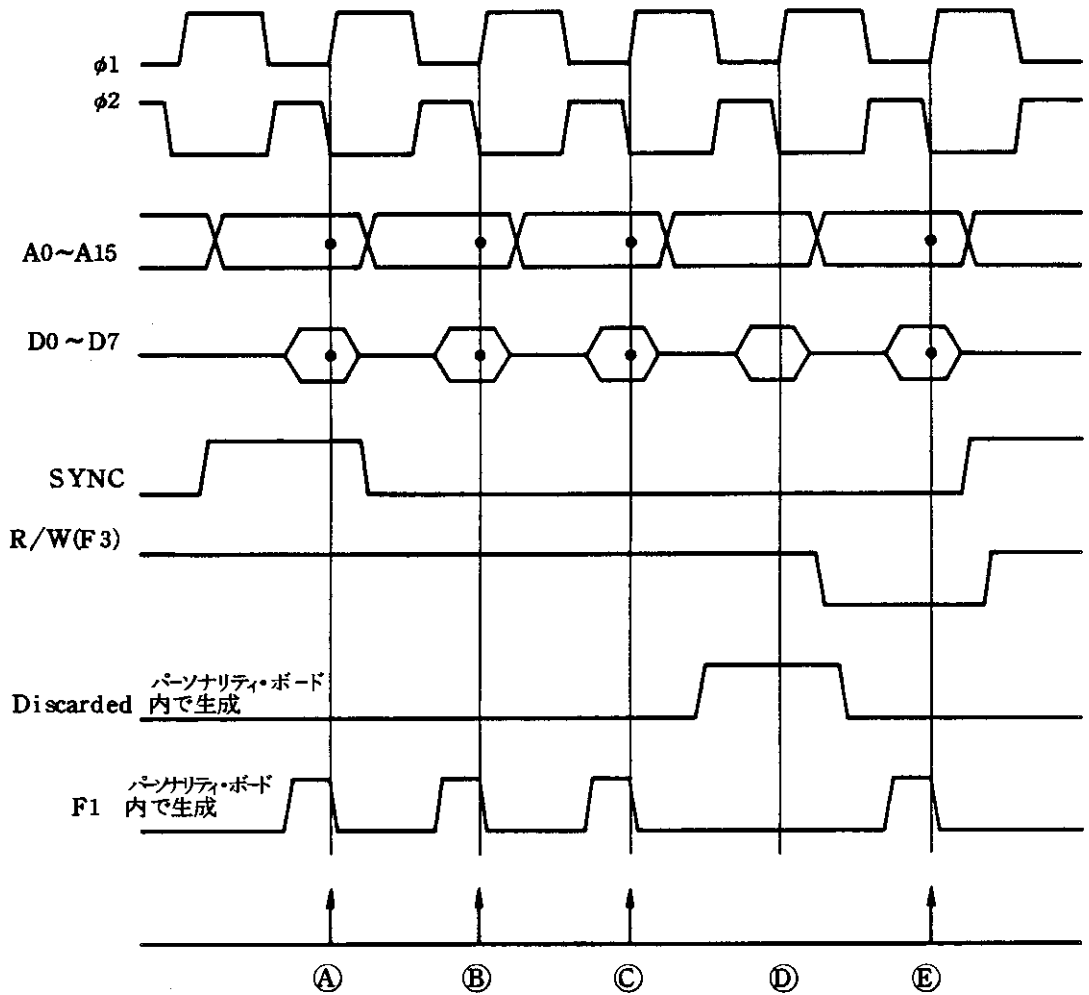
= 0 (R/W信号がロー;ライト・サイクル)

F 2, F 4 は使用されません。

以上のデータは  $\mu$ P のコントロール信号から作られるサンプリング・クロックによってサンプルされますが、そのタイミングを〔図-8〕に示します。

例：

$\textcircled{A}$                      $\textcircled{C}$     $\textcircled{B}$   
 STA                    0 1 0 0, X  
 $\textcircled{D}$                     / RD                    Discarded Data なので表示されない。  
 $\textcircled{B}$   
 0 1                    / WR



$\textcircled{D}$ はDiscarded Dataのためサンプルされない。

図-8 パーソナリティ・ボード内での  
データ・サンプリング・タイミング

サンプルされないサイクルは、[表2※]で○印をつけたサイクルです。

なお、実行アドレスが他のページにまたがる場合、そのサイクルのデータは不要データとなりますが、このデータはサンプルし、表示します。

※ ロックウェル・インターナショナル社発行「R6500 MICRO COMPUTER SYSTEM HARDWARE MANUAL」より引用。

A. 1. SINGLE-BYTE INSTRUCTIONS

ASL	DEX	NOP	TAX	TYA
CLC	DEY	ROL	TAY	
		ROR		
CLD	INX	SEC	TSX	
CLI	INY	SED	TXA	
CLV	LSR	SEI	TXS	

These single-byte instructions require two cycles to execute. During the second cycle the address of the next instruction in program sequence will be placed on the address bus. However, the OP CODE which appears on the data bus during the second cycle will be ignored. This same instruction will be fetched on the following cycle, at which time it will be decoded and executed. The ASL, ROL and LSR instructions apply to the accumulator mode of address.

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
(T1)	PC + 1	OP CODE (Discarded)	1	
T0	PC + 1	OP CODE	1	Next Instruction

A. 2. INTERNAL EXECUTION ON MEMORY DATA

ADC	CMP	EOR	LDY
AND	CPX	LDA	ORA
BIT	CPY	LDX	SEC

The instructions listed above will execute by performing operations inside the microprocessor using data fetched from the effective address. This total operation requires three steps. The first step (one cycle) is the OP CODE fetch. The second (zero to four cycles) is the calculation of an effective address. The final step is the fetching of the data from the effective address. Execution of the instruction takes place during the fetching and decoding of the next instruction.

表2 ○印のサイクルは、不要データですので、サンプルされません。

This section contains an outline of the data on both the address bus and the data bus for each cycle of the various processor instructions. It tells the system designer exactly what to expect while single-cycling through a program.

Note that the processor will not stop in any cycle where R/W is a 0 (WRITE cycle). Instead, it will go right into the next READ cycle and stop there. For this reason, some instructions may appear to be shorter than indicated here.

All instructions begin with T0 and the fetch of the OP CODE and continue through the required number of cycles until the next T0 and the fetch of the next OP CODE.

While the basic terminology used in this appendix is discussed in the Programming Manual, it has been defined below for ease of reference while studying Single-Cycle Execution.

OP CODE--The first byte of the instruction containing the operator and mode of address.

OPERAND--The data on which the operation specified in the OP CODE is performed. BASE ADDRESS--The address in Indexed addressing modes which specifies the location in memory to which indexing is referenced. The high-order byte of the base address (AB08 to AB15) is BAH (Base Address High) and the low-order byte of the base address (AB00 to AB07) is BAL (Base Address Low).

EFFECTIVE ADDRESS--The destination in memory in which data are to be found. The effective address may be loaded directly as in the case of Page Zero and Absolute Addressing or may be calculated as in Indexing operations. The high-order byte of the effective address (AB08 to AB15) is ADH and the low-order byte of the effective address (AB00 to AB07) is ADL.

INDIRECT ADDRESS--The address found in the operand of instructions utilizing (Indirect), Y which contains the low-order byte of the base address. IAH and IAL represent the high- and low-order bytes.

JUMP ADDRESS--The value to be loaded into Program Counter as a result of a Jump instruction.

**A. 2.5. Absolute, X or Absolute, Y Addressing (4 or 5 Cycles)**

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Low-Order Byte of Base Address
T2	PC + 2	BAH	1	Fetch High-Order Byte of Base Address
T3	ADL: BAL + Index Register ADH: BAH + C	Data* Data	1	Fetch Data (No Page Crossing) Carry is 0 or 1 as Required from Previous Add Operation
T4*	ADL: BAL + Index Register ADH: BAH + 1	Data Data	1	Fetch Data from Next Page
T0	PC + 3	OP CODE	1	Next Instruction

\*If the page boundary is crossed in the indexing operation, the data fetched in T3 is ignored. If page boundary is not crossed, the T4 cycle is bypassed.

**A. 2.6. Zero Page, X or Zero Page, Y Addressing Modes (4 Cycles)**

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Page Zero Base Address
T2	00, BAL	Data (Discarded)	1	
T3	00, BAL + Index Register	Data	1	Fetch Data (No Page Crossing)
T0	PC + 2	OP CODE	1	Next Instruction

**A. 2.1. Immediate Addressing (2 Cycles)**

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	Data	1	Fetch Data
T0	PC + 2	OP CODE	1	Next Instruction

**A. 2.2. Zero Page Addressing (3 Cycles)**

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch Effective Address
T2	00, ADL	Data	1	Fetch Data
T0	PC + 2	OP CODE	1	Next Instruction

**A. 2.3. Absolute Addressing (4 Cycles)**

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch Low-Order Effective Address Byte
T2	PC + 2	ADH	1	Fetch High-Order Effective Address Byte
T3	ADH, ADL	Data	1	Fetch Data
T0	PC + 3	OP CODE	1	Next Instruction

**A. 2.4. Indirect, X Addressing (6 Cycles)**

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Page Zero Base Address
T2	00, BAL	Data (Discarded)	1	
T3	00, BAL + X	ADL	1	Fetch Low-Order Byte of Effective Address
T4	00, BAL + X + 1	ADH	1	Fetch High-Order Byte of Effective Address
T5	ADH, ADL	Data	1	Fetch Data
T0	PC + 2	OP CODE	1	Next Instruction

A. 2.7. Indirect, Y Addressing Mode (5 or 6 Cycles)

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	IAL	1	Fetch Page Zero Indirect Address
T2	00, IAL	BAL	1	Fetch Low-Order Byte of Base Address
T3	00, IAL + 1	BAH	1	Fetch High-Order Byte of Base Address
T4	ADL: BAL + Y Data* ADH: BAH + C		1	Fetch Data from Same Page Carry is 0 to 1 as Required from Previous Address Operation
T5*	ADL: BAL + Y Data ADH: BAH + 1		1	Fetch Data from Next Page

T0 PC + 2 OP CODE 1 Next Instruction  
\*If page boundary is crossed in indexing operation, the data fetch in T4 is ignored. If page boundary is not crossed, the T5 cycle is bypassed.

A. 3. STORE OPERATIONS

STA  
STX  
STY

The specific steps taken in the Store Operations are very similar to those taken in the previous group (internal execution on memory data). However, in the Store Operation, the fetch of data is replaced by a WRITE (R/W = 0) cycle. No overlapping occurs and no shortening of the instruction time occurs on indexing operations.

A. 3.1. Zero Page Addressing (3 Cycles)

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch Zero Page Effective Address
T2	00, ADL	Data	0	Write Internal Register to Memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 3.2. Absolute Addressing (4 Cycles)

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch Low-Order Byte of Effective Address
T2	PC + 2	ADH	1	Fetch High-Order Byte of Effective Address
T3	ADH, ADL	Data	0	Write Internal Register to Memory
T0	PC + 3	OP CODE	1	Next Instruction

A. 3.3. Indirect, X Addressing (6 Cycles)

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Page Zero Base Address
T2	00, BAL	Data (Discarded)	1	
T3	00, BAL + X	ADL	1	Fetch Low-Order Byte of Effective Address
T4	00, BAL + X + 1	ADH	1	Fetch High-Order Byte of Effective Address
T5	ADH, ADL	Data	0	Write Internal Register to Memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 3.4. Absolute, X or Absolute, Y Addressing (5 Cycles)

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Low-Order Byte of Base Address
T2	PC + 2	BAH	1	Fetch High-Order Byte of Base Address
T3	ADL: BAL + Index Register ADH: BAH + C	Data (Discarded)	1	
T4	ADH, ADL	Data	0	Write Internal Register to Memory
T0	PC + 3	OP CODE	1	Next Instruction

A. 3.5. Zero Page, X or Zero Page, Y Addressing Modes (4 Cycles)

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Page Zero Base Address
T2	00, BAL	Data (Discarded)	1	
T3	ADL: BAL + Index Register	Data	0	Write Internal Register to Memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 4.1. Zero Page Addressing (5 Cycles)

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch Page Zero Effective Address
T2	00, ADL	Data	1	Fetch Data
T3	00, ADL	Data	0	
T4	00, ADL	Modified Data	0	Write Modified Data Back into Memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 3.6. Indirect, Y Addressing Mode (6 Cycles)

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	IAL	1	Fetch Page Zero Indirect Address
T2	00, IAL	BAL	1	Fetch Low-Order Byte of Base Address
T3	00, IAL + 1	BAH	1	Fetch High-Order Byte of Base Address
T4	ADL: BAL + Y Data (Discarded)		1	
T5	ADH: BAH		0	Write Internal Register to Memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 4.2. Absolute Addressing (6 Cycles)

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch Low-Order Byte of Effective Address
T2	PC + 2	ADH	1	Fetch High-Order Byte of Effective Address
T3	ADH, ADL	Data	1	
T4	ADH, ADL	Data	0	
T5	ADH, ADL	Modified Data	0	Write Modified Data Back into Memory
T0	PC + 3	OP CODE	1	Next Instruction

A. 4.3. Zero Page, X Addressing (6 Cycles)

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Page Zero Base Address
T2	00, BAL	Data (Discarded)	1	
T3	ADL: BAL + X Data (Without Carry)		1	Fetch Data
T4	ADL: BAL + X Data (Without Carry)		0	
T5	ADL: BAL + X Modified Data (Without Carry)		0	Write Modified Data Back into Memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 4. READ -- MODIFY -- WRITE OPERATIONS

ASL  
 DEC  
 INC  
 LSR  
 ROL  
 ROR

The Read -- Modify -- Write operations involve the loading of operands from the operand address, modification of the operand and the resulting modified data being stored in the original location.

A. 4.4. Absolute, X Addressing (7 Cycles)

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Low-Order Byte of Base Address
T2	PC + 2	BAH	1	Fetch High-Order Byte of Base Address
(T3)	ADL: BAL + X ADH: BAH + C	Data (Discarded)	1	
T4	ADL: BAL + X ADH: BAH + C	Data	1	Fetch Data
(T5)	ADH, ADL	Data	0	
T6	ADH, ADL	Modified Data	0	Write Modified Data Back into Memory
T0	PC + 3	OP CODE	1	New Instruction

A. 5. MISCELLANEOUS OPERATIONS

BCC	BRK	PHP
BCS	BVC	PLA
BEQ	BVS	PLP
BMI	JMP	RTI
BNE	JSR	RTS
BPL	PHA	

A. 5.1. Push Operation -- PHP, PHA (3 Cycles)

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
(T1)	PC + 1	OP CODE (Discarded)	1	
T2	Stack Pointer* Data	Data	0	Write Internal Register into Stack
T0	PC + 1	OP CODE	1	Next Instruction

\*Hereafter referred to as "Stack Ptr."

A. 5.2. Full Operations -- PIP, PLA (4 Cycles)

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
(T1)	PC + 1	OP CODE (Discarded)	1	
(T2)	Stack Ptr.	Data (Discarded)	1	
T3	Stack Ptr. + 1	Data	1	Fetch Data from Stack
T0	PC + 1	OP CODE	1	Next Instruction

A. 5.3. Jump to Subroutine -- JSR (6 Cycles)

In	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch Low-Order Byte of Subroutine Address
(T2)	Stack Ptr.	Data (Discarded)	1	
T3	Stack Ptr.	PCH	0	Push High-Order Byte of Program Counter to Stack
T4	Stack Ptr. - 1	PCL	0	Push Low-Order Byte of Program Counter to Stack
T5	PC + 2	ADH	1	Fetch High-Order Byte of Subroutine Address
T0	Subroutine Address (ADH, ADL)	OP CODE	1	

A. 5.4. Break Operation -- (Hardware Interrupt)--BRK (7 Cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch BRK OP CODE (or Force BRK)
⊛(T1)	PC + 1 (PC on Hardware Interrupt)	Data (Discarded)	1	
T2	Stack Ptr.	PCH	0	Push High-Order Byte of Program Counter to Stack
T3	Stack Ptr. - 1	PCL	0	Push Low-Order Byte of Program Counter to Stack
T4	Stack Ptr. - 2	P	0	Push Status Register to Stack
T5	FFFF (NMI-FFFA) (RES-FFFC)	ADL	1	Fetch Low-Order Byte of Interrupt Vector
T6	FFFF (NMI-FFFB) (RES-FFFD)	ADH	1	Fetch High-Order Byte of Interrupt Vector
T0	Interrupt Vector (ADH, ADL)	OP CODE	1	Next Instruction

A. 5.5. Return from Interrupt -- RTI (6 Cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
⊛(T1)	PC + 1	Data (Discarded)	1	
⊛(T2)	Stack Ptr.	Data (Discarded)	1	
T3	Stack Ptr. + 1	Data	1	Pull P from Stack
T4	Stack Ptr. + 2	Data	1	Pull PCL from Stack
T5	Stack Ptr. + 3	Data	1	Pull PCH from Stack
T0	PCH, PCL	OP CODE	1	Next Instruction

※ Hardware Interrupt の場合は、サンプルされます。

A. 5.6. Jump Operation -- JMP

A.5.6.1. Absolute Addressing Mode (3 Cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch Low-Order Byte of Jump Address
T2	PC + 2	ADH	1	Fetch High-Order Byte of Jump Address
T0	ADH, ADL	OP CODE	1	Next Instruction

A.5.6.2. Indirect Addressing Mode (5 Cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	IAL	1	Fetch Low-Order Byte of Indirect Address
T2	PC + 2	IAH	1	Fetch High-Order Byte of Indirect Address
T3	IAH, IAL	ADL	1	Fetch Low-Order Byte of Jump Address
T4	IAH, IAL + 1	ADH	1	Fetch High-Order Byte of Jump Address
T0	ADH, ADL	OP CODE	1	Next Instruction

A. 5.7. Return from Subroutine -- RTS (6 Cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
⊛(T1)	PC + 1	Data (Discarded)	1	
⊛(T2)	Stack Ptr.	Data (Discarded)	1	
T3	Stack Ptr. + 1	PCL	1	Pull PCL from Stack
T4	Stack Ptr. + 2	PCH	1	Pull PCH from Stack
⊛(T5)	PCH, PCL (from Stack)	Data (Discarded)	1	
T0	PCH, PCL + 1	OP CODE	1	Next Instruction



A. 5.8. Branch Operation -- BCC, BCS, BEQ, BMI, BNE, BPL, BVC, BVS (2, 3, or 4 Cycles)

Th	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	Offset	1	Fetch Branch Offset
(T2)	PC + 2 + Offset (w/o Carry)	OP CODE	1	Offset Added to Program Counter
(T3)	PC + 2 + Offset (with Carry)	OP CODE	1	Carry Added

\*Skip if branch not taken.

\*\*Skip if branch not taken; skip if branch operation does not cross page boundary.

4-2 取込まれたデータの表示に対するコメント

4-2-1 逆アセンブル・フォーマットについて

1. オペコードのニーモニックは、ROCKWELL社の標準アセンブリ・フォーマットを使用しています。

(資料: R6500 MICRO COMPUTER SYSTEM PROGRAMING MANUAL)

2. オペランド・データは、すべて16進で表示します。

なお、数値ベースを示します\$記号は使用していません。

例: **LDA +\$10**  $\longleftrightarrow$  **LDA +10**

(ROCKWELL社フォーマット) (TR4720フォーマット)

3. オペコードがあつて、かつオペランドが「データ・メモリ」にない場合は、オペランドの表示は「+++」となります。

例: 「データ・メモリ」の最後のステートがアブソリュート表示で**A9 / OP**の場合は、ニーモニック表示では「**LDA ++++**」となります。

4. リラティブ・アドレッシング・モードの場合は、オペランドは1バイトですが、内部で16進4桁の絶対アドレスに変換して表示します。

例: **BEQ &1234**

&はリラティブ・アドレッシングを示す。

5. JSR オペコード時の表示方法

JSR オペコードをサンプルした場合、そのニーモニック表示は他のオペコードと表示方法が異なります。

例:	[ADRS]	[DATA]	[ADRS]	
	<b>F940</b>	<b>20 / OP</b>	<b>F940</b>	<b>JSR</b>
		<sup>※2</sup> <b>A0</b> / RD		<b>++</b> <span style="border: 1px solid black; padding: 2px;"><sup>※2</sup><b>A0</b></span>
	<b>F941</b>	<b>A0</b> / RD		
	<b>01FF</b>	<b>F9 / WR</b>	<b>01FF</b>	<b>F9</b> / WR
	<b>01FE</b>	<b>42 / WR</b>	<b>01FE</b>	<b>42</b> / WR
	<b>F942</b>	<sup>※1</sup> <b>F9</b> / RD	<b>F942</b>	<sup>※1</sup> <span style="border: 1px solid black; padding: 2px;"><b>F9</b></span> / RD
	<b>F9A0</b>	<b>A5 / OP</b>	<b>F9A0</b>	<b>LDA</b> ++

(アブソリュート表示)                      [ニーモニック表示]

ジャンプ先のアドレスは、(※1)+(※2)=F9A0となります。

4-2-2  $\mu$ Pの特別な動作における表示について

1. R6500の命令セット中の Future Expansion オペコードが発見された場合には、「ノーマル・プリンク表示」の「?OP」で表示します。

例：            **04/OP**                            **04 ?OP**  
                   (アブソリュート表示)            (ニーモニック表示)

なお、一度「?OP」が表示されますと、それ以後の表示は正規のプログラムとは全く異なったプログラム表示となり、 $\mu$ Pの動作自体も異なりますので注意して下さい。これは、 $\mu$ P自体が Future Expansion オペコードに出会いますと、不定の数サイクルの間アドレスをアクセスし、リード、ライトなどの不特定な動作をします。したがって、次にオペコードがあっても単なるリード・データとして扱われたりして、これ以後 $\mu$ Pは全く別のインストラクションを実行することになります。

例：	<b>68 /OP</b>	<b>PLA</b>	}	※	<b>04 /OP</b>	※	<b>04 ?OP</b>
	<b>02 /RD</b>	<b>02/RD</b>			<b>02 /RD</b>	<b>02 /RD</b>	
	<b>A2 /OP</b>	<b>LDX#05</b>		⇒	<b>A2 /RD</b>	<b>A2 /RD</b>	
	<b>05 /RD</b>				<b>05 /OP</b>	<b>ORA A0</b>	
	<b>A0 /OP</b>	<b>LDY#01</b>			<b>A0 /RD</b>		
	<b>01 /RD</b>				<b>01 /OP</b>	<b>ORA (+++</b>	

(アブソリュート表示)(ニーモニック表示)            (アブソリュート表示)(ニーモニック表示)

正規のプログラム

※ Future Expansion オペコードが入りますと、それ以後 $\mu$ Pは右側のプログラムを実行しています。



## 第5章 動作チェック

CPU プロブ先端のデータ取込み点から、本体内部の「データ・メモリ」までの間に、コネクタや IC などによる多くの接続点があります。そのため、常に正しいデータの取込みが行なえるように、パーソナリティ・ボードには  $\mu$ P のバスおよびコントロール信号に準じたテスト・パターン発生器があり、背面パネルの **CPU PROBE TEST** ソケットへテスト・パターンが出力されています。このテスト・パターンによってチェックしますと、プロブ先端から「データ・メモリ」までのルートが正常に動作しているかどうか確認することができます。以下に示す手順でチェックを行なって下さい。

1. **CPU PROBE** に、40 ピン DIP クリップ・コネクタあるいは 40 ピン DIP プラグ・コネクタを接続します。
2. 背面パネルの **CPU PROBE TEST** ソケットへは、プラグ・コネクタの場合は直接に、クリップ・コネクタの場合は付属の「40 ピン DIP IC パッケージ」を取付けてから接続して下さい。
3. **TRACE** キーによって「**TRACE STATE ALL**」モードに設定します。  
次に **DEFAULT** キーを押して、すべての「トレース条件」を初期状態として下さい。
4. 「入力プロンプト」が「**TRIG[ADRS]**」の位置にありますから **ENTRY** キーによって **0000** と入力します。
5. **RUN** キーを押しますと〔写真5〕に示すようなテスト・パターンが CRT ディスプレイに表示されます。アドレス・バス・データとデータ・バス・データの各桁が同一の数字となっていることを確認して下さい。
6. 必要に応じて他のファンクションの基本動作の確認が行なえます。



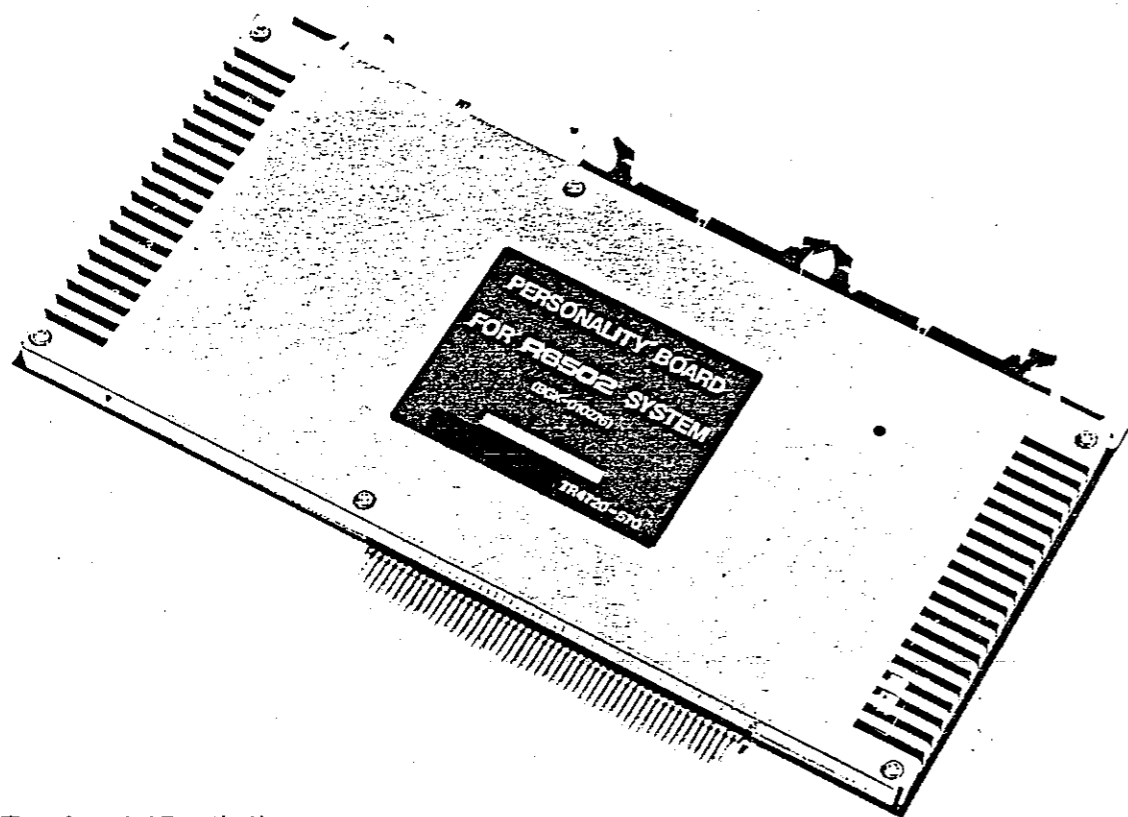


写真1 パーソナリティ・ボード

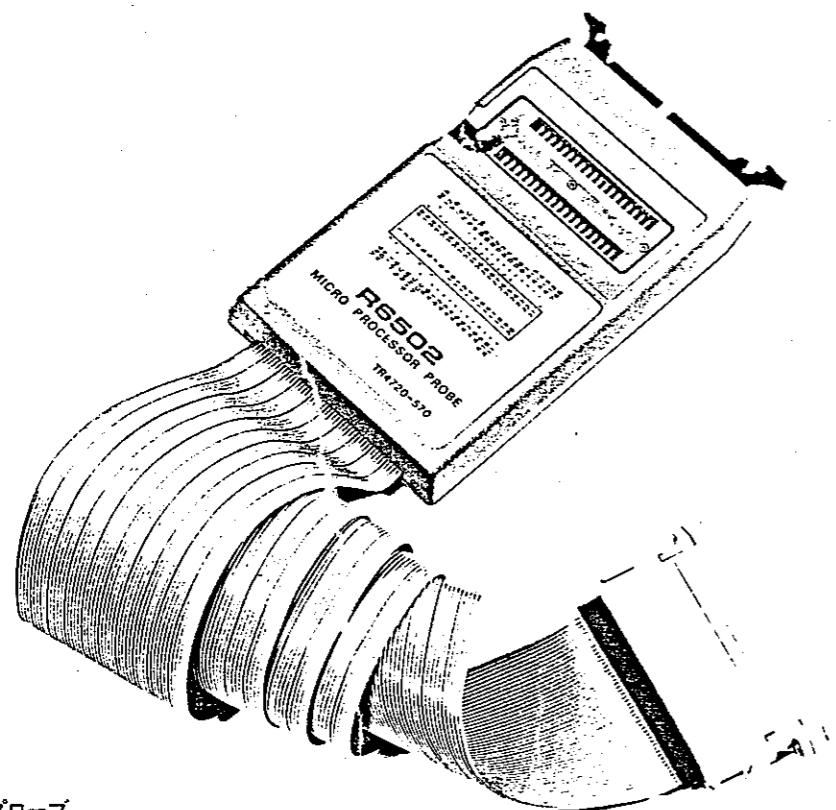
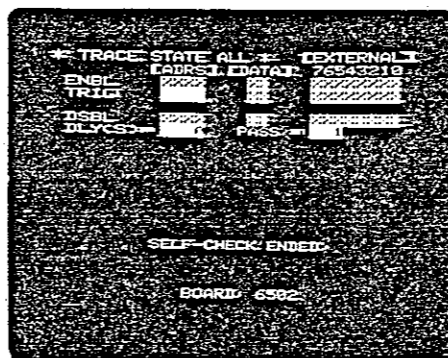


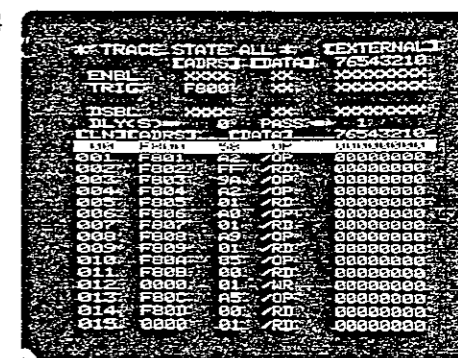
写真2 CPUプローブ

写真3



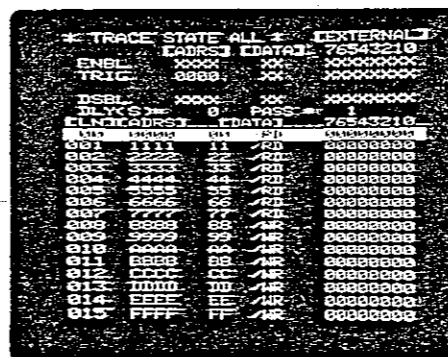
POWER ON時のパーソナリティ・ボード  
名称表示

写真4



「TRACE STATE ALL」  
ABSモードの表示

写真5



テスト・パターン







## 本製品に含まれるソフトウェアのご使用について

本製品に含まれるソフトウェア（以下本ソフトウェア）のご使用について以下のことにご注意下さい。

ここでいうソフトウェアには、本製品に含まれる又は共に使用されるコンピュータ・プログラム、将来弊社よりお客様に提供されることのある追加、変更、修正プログラムおよびアップデート版のコンピュータ・プログラム、ならびに本製品に関する取扱説明書等の付随資料を含みます。

### 使用許諾

本ソフトウェアの著作権を含む一切の権利は弊社に帰属いたします。

弊社は、本ソフトウェアを本製品上または本製品とともに使用する限りにおいて、お客様に使用を許諾するものといたします。

### 禁止事項

お客様は、本ソフトウェアのご使用に際し以下の事項は行わないで下さい。

- 本製品使用目的以外で使用する事
- 許可なく複製、修正、改変を行う事
- リバース・エンジニアリング、逆コンパイル、逆アセンブルなどを行う事

### 免責

お客様が、本製品を通常の用法以外の用法で使用したことにより本製品に不具合が発生した場合、およびお客様と第三者との間で著作権等に関する紛争が発生した場合、弊社は一切の責任を負いかねますのでご了承下さい。

# 保証について

製品の保証期間は、お客様と別段の取り決めがある場合または当社が特に指定した場合を除き、製品の納入日(システム機器については検取日)から1年間といたします。保証期間中に、当社の責めに帰する製造上の欠陥により製品が故障した場合、無償で修理いたします。ただし、下記に該当する場合は、保証期間中であっても保証の対象から除外させていただきます。

- 当社が認めていない改造または修理を行った場合
- 支給品等当社指定品以外の部品を使用した場合
- 取扱説明書に記載する使用条件を超えて製品を使用した場合(定められた許容範囲を超える物理的ストレスまたは電流電圧がかかった場合など)
- 通常想定される使用環境以外で製品を使用した場合(腐食性の強いガス、塵埃の多い環境等による電気回路の腐食、部品の劣化が早められた場合など)
- 取扱説明書または各種製品マニュアルの指示事項に従わずに使用された場合
- 不注意または不当な取扱により不具合が生じた場合
- お客様のご指示に起因する場合
- 消耗品や消耗材料に基づく場合
- 火災、天変地異等の不可抗力による場合
- 日本国外に持出された場合
- 製品を使用できなかったことによる損失および逸失利益

当社の製品の保証は、本取扱説明書に記載する内容に限られるものとします。

## 保守に関するお問い合わせについて

長期間にわたる信頼性の保証、国家標準とのトレーサビリティを実現するためにアドバンテストでは、工場から出荷された製品の保守に対し、カスタマ・エンジニアを配置しています。

カスタマ・エンジニアは、故障などの不慮の事故は元より、製品の長期間にわたる性能の保証活動にフィールド・エンジニアとしても活動しています。

万一、動作不良などの故障が発生した場合には、当社のMS(計測器)コールセンターにご連絡下さい。

## 製品修理サービス

- 製品修理期間  
製品の修理サービス期間は、製品の納入後10年間とさせていただきます。
- 製品修理活動  
当社の製品に故障が発生した場合、当社に送っていただく引取り修理、または当社技術員が現地に出張しての出張修理にて対応いたします。

## 製品校正サービス

- 校正サービス  
ご使用中の製品に対し、品質および信頼性の維持を図ることを目的に行うもので、校正後の製品には校正ラベルを貼付けし、品質を保証いたします。
- 校正サービス活動  
校正サービス活動は、株式会社アドバンテスト カスタマサポートに送っていただく引取り校正、または当社技術員が現地に出張しての出張校正にて対応いたします。

## 予防保守のおすすめ

製品にはエレクトロニクス部品およびメカニカル部品の一部に寿命を考慮すべき部品を使用しているため、定期的な交換を必要とします。適正な交換期間を過ぎて使用し発生した障害に対しては、修理および性能の保証ができません場合があります。

アドバンテストでは、このようなトラブルを未然に防ぐため、予防保守が有効な手段と考え、予防保守作業を実施する体制を整えています。

各種の予防保守を定期的実施することで、製品の安定稼働を図り、不意の費用発生を防ぐため、年間保守契約による予防保守の実施をお勧めいたします。

なお、年間保守契約は、製品、使用状況および使用環境により内容が変わりますので、最寄りの弊社営業支店にお問い合わせ下さい。

# ADVANTEST

<http://www.advantest.co.jp>

## 株式会社アドバンテスト

本社事務所  
〒100-0005 千代田区丸の内1-6-2 新丸の内センタービルディング  
TEL: 03-3214-7500 (代)

第4アカウント販売部(東日本)  
〒100-0005 千代田区丸の内1-6-2 新丸の内センタービルディング  
TEL: 0120-988-971  
FAX: 0120-988-973

第4アカウント販売部(西日本)  
〒564-0062 吹田市垂水町3-34-1  
TEL: 0120-638-557  
FAX: 0120-638-568

### ★計測器に関するお問い合わせ先

(製品の仕様、取扱い、修理・校正等計測器関連全般)

MS(計測器)コールセンタ ☎ TEL 0120-919-570  
FAX 0120-057-508

E-mail: [icc@acs.advantest.co.jp](mailto:icc@acs.advantest.co.jp)