

微处理器测试图案产生方法研究

李 云 岗 林 雨
(中国科学院半导体研究所)

1984年4月30日收到

本文提出一种微处理器测试程序的编制方法——T 分类法，并运用这种方法编制了 8080 微处理器测试程序，并将部份程序进行了测试。与指令功能分类法相比，应用此法可以节省图案贮存器容量 78.5%，适用范围广，有利于故障分析。

本文还介绍了利用建立标准图案单元库的方法在 MDR-Z80 微型机上实现的半自动图案生成。并探索了全自动图案生成的可能性。

一、引言

微处理器是由成千上万个门电路构成的复杂功能系统，并被集成在一个有限引脚的芯片中。这就给微处理器测试带来了不同于其它电路的特殊困难和新的要求。微处理器测试方法的研究一直是国内外都很关心的一个课题^[4,6-10]。

微处理器功能测试必须解决测试指令序列的安排和测试图案生成二个问题。

目前见到的依据用户信息确定测试序列的主要有硬件模块法^[6] 和指令功能分类法。模块法是 1976 年出现的一种方法，它是将微处理器从硬件结构上划分为一系列模块，例 ALU、累加器、PC、SP 等。对每一个模块来说都可以通过执行一组指令，直接或间接地与 I/O 总线相联，从而把测试结果送到外部进行比较。将依次检验每一模块的指令连接起来就形成总的测试程序。模块法出现的比较早，无疑对当时微处理器测试起到了推动作用，但由于当时认识的局限，其中也有一些不足。后来，又有人对模块法提出了改进，例如 D. H. Smith 提出的 WISEST 方法等。

另外，任何一个微处理器的指令系统根据其功能都可以分为数传指令、算术运算指令、逻辑运算指令等。依次对每一类指令进行检验，也可形成总的测试程序，这就是所谓的指令功能分类法。日本港电子 9200 测试系统中的 8080 微处理器测试程序就是依此法编制的。它检验了 8080 微处理器的 72 条基本指令、244 种操作码。

测试程序的编制方法还有美国伊里诺思大学 S. M. Thatte 提出的算法和法国 C. Robach 提出的按指令复杂程度分类的方法^[7-10]。

微处理器的测试图案由指令执行过程中每节拍地址总线、数据总线和控制总线上的信息构成。按照测试图案形成的方法又有学习法、存贮响应法、仿真法、算法产生法和功能模拟比较法等之分。

功能模拟比较法是在总结各种测试方法的基础上于 1977 年提出的一种测试方法^[1,2]。并以此法为基础先后设计了几台测试仪。其中比较典型的有 2000A 大规模集成

电路测试系统。2000A 测试系统的图案发生器不仅可以用算法产生存贮器的测试图案和用存贮响应法产生随机逻辑电路的测试图案，并且还能够用功能模拟比较法产生复杂的大规模集成电路的测试图案。由于它是用高速专用 CPU 通过指令运算来修改图案存贮器中的部份图案，然后再将修改后的图案高速推出。因此，用功能模拟比较法产生微处理器的测试图案可以节省图案存贮器容量，减少图案生成程序。但同时也提出一个问题，即测试指令序列如何编排才能更好地发挥这种图案发生器的效率。这正是本文所要解决的问题之一。本文所要解决的另一问题是如何生成测试图案。

二、T 分类法

由于微处理器集成度高、引脚少、可观测点极其有限，给按结构产生算法带来了困难。另外，由于微处理器的全部功能均由其指令系统实现，每一条指令按时钟周期展开的全部微操作可在生产厂家提供的器件手册中找到，因此微处理器测试可以通过单指令检验生成微处理器的测试图案。指令序列的安排应考虑二个问题：一能对微处理器的各种功能逐次检验；二使得测试图案尽量短。依据这二条原则，本文提出依指令所含有的时钟周期数，即 T 状态数把指令进行分类的方法进行测试程序的编排。为了方便，以下均将此法称为 T 分类法。

指令一般由机器周期组成，机器周期又含有数量不等的时钟周期。时钟周期也叫 T 状态，或称节拍。我们依每条指令的 T 状态数作为该条指令的“权”。例如具有 4 个 T 状态的指令，就称为 4T 指令。依每条指令的“权”，将指令分类，具有 n 个 T 状态就称为 n T 类指令。

定义 1：输入

我们仅考虑可访问的微处理器内部寄存器，数据可以从外部传送到这些寄存器。所谓外部是指存贮器或输入设备。这个外部相对于微处理器内部寄存器而言，就是一个输入，或称控制点。

定义 2：输出

数据经微处理器内部处理之后，必须传送到外部，即总线上，以便检测、比较。它相对于微处理器内部寄存器而言，就是一个输出，或称观测点。

上面虽然二次用到外部，但其含义是不同的。上述思想可以表示如图 1。

定义 3：写指令

对于传输类指令来说，均有传输的源和传输的目的。凡传输源为输入，目的为微处理器内部寄存器的传输类指令为写指令。

定义 4：读指令

传输源为微处理器内部寄存器，目的输出的传输类指令为读指令。

规则 1：测试序列

测试序列是一个从输入到输出的有向数据流通路。它至少应由一条写指令和一条读指令组成。如图 2 所示。

规则 2：在一个测试序列中只安排一条未经测试的指令，该测试序列就叫做这条未

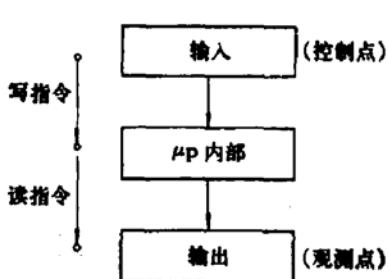


图 1

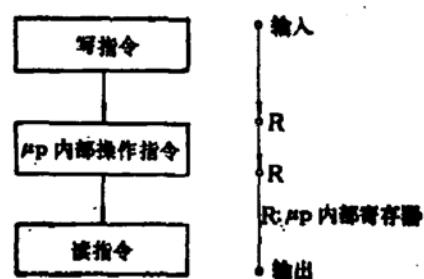


图 2

经测试的指令的检测序列。例如, 测试序列 A 中只有一条“ADD B”指令是待测指令, 那么测试序列 A 就叫做指令“ADD B”的测试序列。

规则 3: 为了缩短测试图案, 组成测试序列时, 在满足检测要求的前提下, 应尽量采用“权”比较小的写指令。

规则 4: 在安排测试序列前后次序时, 应把 T 状态数相同且操作码相邻的指令的检测序列尽量安排在一起, 以便于生成测试图案。

规则 5: 在测试序列次序的安排中, 应尽量使前面检测过的指令为后面待检指令服务, 以便于故障分析。

规则 6: 开始时, 没有一条指令是检验过的, 为避免用未经检测过的指令去检测其它未经检测过的指令的状况, 所以应选择一个最小指令集, 用极少数写指令和读指令构成一个可以互相校检的序列。如果其中有一条指令功能不正确, 则可由其它指令检测出来。待确认这部份指令功能正确之后, 再由它们和其它待检指令构成测试序列。

规则 7: 测试数据的选择

由于不可能对所有操作数的组合情况都检测一遍, 所以应尽量选择一些要求苛刻的操作数进行检验。这样, 既节省了测试时间, 又保证了测试的严格性。

依据上述规则, 我们具体编排了 Intel 8080 微处理器的测试程序, 对其 72 条基本指令, 244 种操作码逐一进行了测试^[5]。由于同一功能指令的 T 状态数相近, 所以 T 分类法和指令功能分类法有类似之处, 但也有其不同, 下面把按 T 分类法编制的 8080 测试程序和国外某测试仪中按指令功能分类法编制的 8080 测试程序作一比较。详见表 1。

表 1 二种 8080 测试程序比较*

参 数	按功能分类的某测试仪程序	按 T 状态分类的测试程序
指令功能检验能力	好	好
程序长度(字节)	1732	1608
全部程序的 T 状态数	9459	9017
全部测试图案所需存贮容量(字节)	$75672 = 73K + 920$	$72136 = 70K + 456$
实际测试需存的图案存贮容量(字节)	75672	13192
实际测试需存的图案占总图案的比例	100%	18.3%
测试命令总量(字节)	<100	2346
实际测试需存贮容量占全部测试图案的比例	100%	21.5%

* 以上参数均是假定在 2000A 大规模集成电路测试系统测试而得的。

从表 1 中可见这两种测试程序的指令功能检验能力是类似的，二者都对 8080 微处理器的指令系统进行了严格的检验。从占用存贮器容量来看，由于按 T 分类法编制的程序可以适用于按功能模拟比较法设计的测试系统，其图案存贮容量仅占其全部图案所需存贮容量的 18.3%，加上图案发生器中 CPU 命令所占存贮器容量，也仅只有其全部图案所需存贮容量的 21.5%。即采用 T 分类法编制的测试程序有利于节省图案存贮器容量。在按 T 分类法编制的测试程序中，首先要安排一个可以互相检测的测试序列，仅在确认一部份写指令和读指令功能正确之后，再由它们去检测其它指令，这样有利于故障分析。而国外某测试仪中的 8080 微处理器测试程序开始时是由一部分未经检验过的指令去检验另一些未经检验过的指令。这对于故障分析肯定不如前者有利。从适用范围来看，按指令功能分类法编制的测试程序一般仅适用于按存贮响应法设计的测试仪。而按 T 分类法编制的测试程序不仅适用于按存贮响应法设计的测试仪，而且适用于按功能模拟比较法设计的测试仪。

三、测试图案的生成

微处理器的指令周期均可分解成一些典型简单时序的组合，而测试图案是依据时序按节拍形成的。因此每条指令的测试图案都可由一些基本测试图案所组成，所不同的仅是图案中相应于数据总线和地址总线的信息是改变的。这是由于微处理器在执行指令过程中，地址总线上的信息必须不断变化，以指出下一条指令的地址。再就是由于指令的操作码是不同的，指令的操作数是随机的，即数据总线上的信息是不断变化的。因此，我们可以把每种机器周期的测试图案作为标准图案先存入单元库，在生成测试图案时，把测试程序中指令的组合转变为机器周期的组合，再按程序执行的顺序依次从单元库中调出所需机器周期的测试图案，仅改变其地址部分和数据部分，从而形成全部所需测试图案。

依据上述思想，我们具体分析了 8080 微处理器，实现了半自动测试图案生成。所谓半自动是指由人把指令划分为机器周期，并给出每一机器周期地址部分和数据部分的值。8080 微处理器有 10 种机器周期，取址又可分为 4T 和 5T 两种。我们把这 11 种机器周期的测试图案作为标准图案存入单元库内；另外，为了安排等待、保持等不规则状况下的测试图案专门设置了一个临时输入端口。在测试图案生成过程中，上述 12 种情况分别用二个英文字母表示它们，如表 2 所示。在程序执行时，就根据这两个英文字母调用该机器周

表 2 机器周期英文表示

表 示	机器周期	表 示	机器周期
F4	4T 取指	IP	输入
F5	5T 取指	OP	输出
MR	存贮器读	IN	中断
MW	存贮器写	HT	暂停
SR	堆栈读	HI	暂停·中断
SW	堆栈写	SE	自编输入

期的测试图案,然后输入地址、数据部分的值。

2000A LSI 测试系统的图案存贮器宽度为 64 位,分为四段,每段 16 位,如表 3 所示。AM 对应于地址总线,DM 对应于数据总线。在测八位机时,仅使用 DM 的低八位,高八位赋 0 值。SM 对应于微处理器的控制总线。PM 对应于测试仪的控制信息。表 4 中给出了 8080 微处理器的 4T 取指周期的测试图案,用“X”表示的是要修改的信息。可以看出要修改的仅是其很小的一部分。

表 3 2000A LSI 测试系统图案存贮器分配表

AM	DM		SM	PM
16 位	16 位		16 位	16 位
地址总线	H	L	μ P 控制总线	测试仪控制端
	00	数据总线		

表 4 8080 μ P 4 T 取指周期测试图案

	AM	DM	SM	PM
T ₁	0000	00A2	0212	0013
T ₂	X X X X	0000	0112	000C
T ₃	0000	00 X X	0012	0007
T ₄	0000	0000	0012	003F

在每一机器周期的测试图案形成之后,都可以显示、修改。在全部测试图案形成之后,测试命令形成之后都可以显示、修改。详见参考文献[5]。

从前面叙述的测试图案半自动生成中可以知道,它是由人把指令划分为机器周期,并给出相应的地址部分和数据部分的值。这对于操作人员的要求是比较高的。为此,我们探讨了测试图案自动生成的可行性。自动生成即输入测试程序之后,不用人干预,直接生成测试图案。自动生成的关键是:仅输入测试程序,计算机是否可以形成机器周期、地址、数据这三种信息。下面具体分析其可能性。

首先,由指令到机器周期,即输入一条指令,机器自动指出由那几个机器周期组成的。这只要把一张指令构成表送入计算机即可实现。

其次,地址部分和数据部分的值可以用模拟的办法获得。将微处理器的每个寄存器分别用一变量表示,依据用户手册中给出的微操作,按测试程序中指令的输入顺序,模拟每一指令的功能,使得这些变量永远都能反映各寄存器当前的值。在生成测试图案时,根据需要送出地址和数据部分的值。在各寄存器的模拟中,最困难的是标志位的模拟。我们采用逐位消的办法来形成标志位,并在论文[5]中给出了形成标志位的程序。

为了验证测试图案自动生成的方案,我们不仅对标志位生成程序进行了试验,而且还选择了三条指令进行了自动生成。试验结果表明,按此方法生成测试图案是可行的。图 3 是自动生成程序框图。

在自动生成程序中,如采用一般汇编程序的办法,即二遍扫描的办法,肯定是可行的。但程序量大,因此我们采用一遍扫描的办法,在输入测试程序的同时,用软件模拟每一指

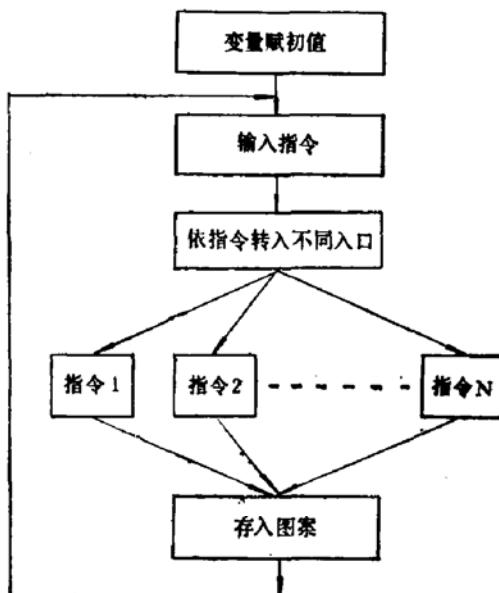


图3 测试图案自动生成框图

令的功能，并生成测试图案。

四、结 论

从整个工作中，我们体会到，功能模拟比较法是一种功能灵活、潜力很大的测试方法。**T** 分类法就是依这种方法为基础而提出的。**T** 分类法的优点是可以节省图案存储器容量，适用范围广。本文提出的以机器周期的测试图案为标准图案，存入单元库，通过逐次调用，仅改变其部分图案的办法生成总的测试图案。可以简化测试图案生成程序。对于不是以机器周期分类的机种，只要把这种思想稍加修改也可以适用。因为微处理器执行指令过程中，其时序关系总是某些标准时序的组合，不可能毫无规律可循，只要抓住其基本时序关系，形成相应的标准图案，就可以建立标准图案单元库。

以上工作是在中国科学院半导体所研制的 2000A LSI 测试系统上做的。工作中得到清华大学陈兆龙教授的帮助，在此表示衷心地感谢。

参 考 文 献

- [1] 林雨，半导体通讯，No. 1, 15—20 (1979).
- [2] 林雨，半导体学报，1, 329—332 (1980.11).
- [3] 陈兆龙，大规模集成电路测量讲义，(1981. 12).
- [4] 陈兆龙等，微处理器测试方法，中国电子学会测量与仪器学会第一届测试会议交流论文(1980).
- [5] 李云岗，中国科学院半导体所硕士论文，(1984.3).
- [6] Chiang, A. C. L and Mccaskill, R., *Electronics*, 49, 100(1976).
- [7] S. M. Thatte and J. A. Abraham, *IEEE Tran. on Computer*, C-29, 429 (1980).
- [8] S. M. Thatte and J. A. Abraham, *FTCS-9*, 203(1979).
- [9] C. Robach and G. Saucier, *EUROCON'80*, 141—144.
- [10] C. Robach and G. Saucier, *IEEE International Test Conference*, 433—443(1980).
- [11] Intel, *The 8080/8085 Microprocessor Book*, (1980).

Research on Microprocessor Test Pattern Generation

Li Yungang and Lin Yü

(Institute of Semiconductors, Academia Sinica)

Abstract

A new method for composing microprocessor test program, the T-classifying method is proposed. This method has been used in composing 8080 μ P test program, of which parts have been tested. Compared with that of instruction function classifying, this method can save 78.5% of the Content of the pattern memory, it has better adaptability and analyses faults more easily.

Semi-automatic pattern generation on MDR-Z80 microcomputer by establishment of standard pattern element storeroom has been realized. The feasibility of fully automatic pattern generation is discussed.