

# 論元器件測試的品質 (I)

■文：羅徹斯特電子 Rochester Electronics 供文

如今，無論是半導體產業的採購、經理人還是採購代理，在提及半導體元器件的電氣 / 參數測試時，都會重點強調一些主要的技術和商業問題。半導體元器件測試的發展歷史與半導體元件本身的歷史同樣久遠，然而除了經驗豐富的工程師之外，其他人對半導體元器件測試瞭解甚少。對於半導體元件的採購和經理人而言，雖然他們認識到測試是採購流程中的必需步驟，但往往並不瞭解具體的測試要求以及測試的價值，因而也就無法針對特定產品、技術或細分市場選擇適合的測試方法。此外，測試服務的有效性非常難以對比，這可能導致對某些元器件匹配並不合適的測試級別。要滿足半導體元器件的測試要求，並採用合適的測試方法，經驗豐富的元器件測試工程師就顯得尤為重要。羅徹斯特電子 (Rochester Electronics) 秉持工程驅動測試的理念，確保在其美國麻塞諸塞紐伯里波特總部的測試工廠中，選擇合適的測試技術和方法對所有制造和待測元器件進行測試。

儘管半導體製造的品質水準在過去十年間已普遍提高，但產業的總體發展越來越需要對半導體元器件進行完整、精確的測試。實際上，隨著半導體元器件越來越複雜，集成度越來越高，拓撲結構越來越簡單，為降低殘餘缺陷率，執行全面測試的需求也越來越高。就測試程式而言，20 世紀 80 年代和 90 年代的測試程式遠比 21 世紀初的測試程式簡單得多；而 21 世紀初測試程式的開發難度遠遠比不上當今的現代測試程式。在今天，現代測試程式可能包含數百萬個測試向量、內建自測試 (BIST) 功能、鐳射或熔絲修調電路、產品分級、測試模式電路和功能等等。這種複雜性 (以及其他因素) 是導致獨立開發人員開發的測試程式品質較低的主導因素。此

類開發人員通常“自我調節”或者自主確定自己的測試方案和測試覆蓋率，測試品質極少受到質疑，也很少經過任何有資質代理機構的審核。

許多 OEM 和製造商都面臨著需要持續降低成本的壓力，他們往往會選擇削減測試成本，來達成降低總體成本的目標，而這種做法會使問題進一步惡化。大多數測試成本削減方法會減少測試過程中使用的資料模式，甚至完全放棄某些資料模式，使得問題元器件僅能依靠抽樣的方式檢測出來。這種降低測試成本的做法通常美其名曰“測試優化”，“最壞情況”測試就是其中之一，這種測試僅檢測元器件在常見參數取特定極值情況下的表現。這些成本削減方法不僅會對半導體器件的品質產生不利影響，還可能導致“異常”批次出廠。這類批次的工作參數值儘管在資料手冊規定的範圍內，但並非典型值，因而異常批次的失效率高於平均水準。異常批次是製造過程中的“漏網之魚”，這類元器件的工作狀況往往在正態曲線之外，從而使系統內部失效的可能性進一步提高或者可能導致系統不穩定。對於應用在個人計算器等消費類產品中的半導體元器件而言，這不會帶來難以承受的風險 / 後果；但對於對可靠性要求高的關鍵應用，後果可能會非常嚴重。

當然，對於一貫表現良好並且有統計和分析資料支撐的特定或給定產品系列，也可以使用優化測試方法；但對於衍生產品或者其他“強化篩選”參數 / 條件，這種優化測試則完全不適用。換言之，如果採用商用元器件的優化測試程式篩選工業用或者軍用元器件，測試品質可能會非常低。如今，在對許多關鍵軍事應用所用的 COTS (現成商用) 元器件進行強化篩選時，這種情況經常出現。

本文旨在讓讀者基本瞭解半導體元器件測試的

基本要素，重點強調測試品質。測試品質對於測試的有效性至關重要，而且並非所有測試的品質均相同。瞭解這些原則可說明負責元器件採購和測試服務的人員選擇合適的測試服務和 / 或測試設備，以滿足元器件測試要求。許多細分市場（商業、汽車、航空航天、軍事等等）都有各自特定的測試要求，而且許多測試技術也都有著不同的故障分類，因此，不熟悉上述原則的人員在選擇測試套件時往往會感到困惑。

解決這一問題的方法是，由經驗豐富且有資質的人員針對測試制定決策，這也是羅徹斯特電子工程驅動測試方案所遵循的理念。這種測試方案最明顯的優勢是，測試工程師完全熟悉各種測試設備，充分掌握待測元器件的功能和工作方式、製造技術以及適用的各種測試方法，對不同細分市場的測試要求了然於胸，並且熟悉元器件適用的故障類別和失效模式。因此，這種方案可以保證羅徹斯特電子所測試的所有元器件，都能夠達到適當的測試覆蓋率。

## “什麼是元器件測試？”

元器件測試是將缺陷元器件與出廠合格元器件區分開來的過程。在這樣的背景下，元器件測試的價值顯而易見。測試過程還是一個無形的過程，因此，很容易被錯誤地解讀和表述。測試在本質上完全是一種分析行為，並不改變或更改被測元器件。關於元器件測試過程的有效性，還有另外一種觀點：

“低品質測試程式總是在嘗試找到一種條件來判定元器件合格，而高品質測試程式則是在尋找任何原因來判斷元器件不合格。”

圖說：典型的測試程序



## “什麼是測試程式？”

測試程式是一組程式碼，用於操作元器件測試平臺上的各種儀器和設備，應用測試條件（激勵信號），然後測定元器件的響應。測得的回應會與用於判斷元器件能否“通過”測試的限值或其他條件進行對比。無論何種元器件系列，所有測試程式都包含三個基本部分：功能測試、參數測試和動態 / 時序測試。

### 功能測試：

測試程式中的功能測試用於確保元器件功能正常。元器件的功能和行為是否符合預期？這很難確定，最容易被錯誤地解讀甚至產生誤導。元器件功能僅能在元器件原始輸入和輸出端進行測定，但測試時必須對所有內部電路進行驗證。而要確定適宜的功能測試數量，通常需要瞭解元器件的產品分類並進行一定程度的故障建模和故障檢測。故障模型分為多種不同的類型，具體可根據被測元器件的類型進行選擇。對於邏輯電路，“固定”故障模型是大多數故障分析工具所採用的主要模型，但這並不表示這種模型涵蓋了可能存在的所有故障類型。

### 參數測試：

測試程式中的參數測試用於保證元器件特性符合其在電路中的預期表現。執行參數測試時，典型的被測參數包括輸出電壓和輸出電流、洩漏電流、供電電流和輸入電壓閾值等等。這些參數是元器件的週邊條件，關乎元器件在電路中的表現。

### 動態 / 時序測試：

測試程式的動態 / 時序測試用於確保元器件的開關速度。功能性決定著元器件的作用，而動態和時序測試則用於確定元器件發揮其作用的速度。對大多數元器件而言，此部分測試都至

關重要，在某些特定的元器件類型（例如，記憶體）中，動態 / 時序故障可能是主要的故障類別。

## “什麼是測試品質？”，“什麼是缺陷器件？”

測試品質是指測試過程從所有元器件中剔除有缺陷元器件的能力。缺陷元器件是指存在可能導致元器件工作不合預期的故障狀況的元器件。總體來看，這些定義和概念都非常簡單，易於理解。但這種簡單僅限於此，要瞭解故障狀況並確定適當的測試覆蓋率則要困難得多，因為故障分為許多類別並且在邏輯上，並非所有故障均為可測故障。在有些情況下，一些故障類別是某些技術所特有的（例如，CMOS 元器件的 IDDQ 故障）。但在許多情況下，故障類別都是產品類別所獨有的（例如，PLD 的熔絲熔斷故障）。

## “什麼是故障？”

故障是元器件中存在的會導致元器件工作與預期不符的一種狀況。邏輯故障是對失效即時影響的一種邏輯抽象，表示無故障電路與有故障電路在元器件一個或多個可觀察原始輸出端表現出的行為差異。要發現元器件故障，必須能夠對元器件施加激勵並且能夠觀察到預期結果，如果測試結果與預期結果不符，將發現故障且元器件不能通過測試。故障僅在邏輯模型中存在可觀察到的邏輯錯誤，並且該錯誤能夠傳播到元器件原始輸出端時才能檢測出來。也就是說，只有當這一與無故障電路截然不同的可觀測值能夠呈現在被測元器件的某個輸出引腳上時，才會認為故障“可測”。電路中還可能存在許多不可測故障，但由於無任何向量序列（輸入激勵）可導致在原始輸出端產生可觀察到的錯誤，這些故障會從目標位置清單中刪除。因此，在故障模型中僅將可測故障作為檢測目標。

## “什麼是故障模型？”

故障模型（通常稱為故障集合）是指一份故障

清單，其中包含作為檢測目標由故障模擬器進行檢測的各種故障。故障模擬器是一種軟體工具，能夠模擬無故障電路在響應輸入激勵（測試向量）時的行為。此外，該模擬器還必須能夠模擬故障條件對邏輯電路行為所產生的影響。

用於識別故障的測試向量基於故障模型而創建。模擬器會逐一檢查故障清單所列的故障，開始生成能夠區分出故障行為與無故障行為的測試序列。對於邏輯電路，這些故障分為不同的類別。

存在於元器件周邊並與輸入輸出引腳直接相關的故障，稱為引腳故障。存在於結構化邏輯模型（指門級元件）輸入和輸出位置的故障，稱為邏輯故障。在饋入其他邏輯故障的公共節點上存在的邏輯故障稱為扇出故障。此類故障來源於扇入和扇出分支，即邏輯門之間的“連線”。

上述故障類別只是故障模型所用的故障類別的一些示例，值得注意的是，故障類別分為許多種，而且並非所有類別對所有類型的元器件和 / 或技術都適用。若要確切瞭解某個器件最理想的故障模型，需要充分瞭解元器件、製造技術以及最終使用者目標應用所需的可靠性級別。

## “什麼是故障覆蓋率？”（不僅僅是覆蓋率，還涉及建模）

由於目標故障數的存在，故障覆蓋率經常會引發困惑，甚至會被錯誤解讀。報告的故障覆蓋率與總故障覆蓋率經常存在差異。故障覆蓋率是指檢測出的故障數與目標檢測故障數的數學比值。由於故障覆蓋率不僅受檢測出的故障數的影響，而且還受到目標故障數的影響，因而非常容易被錯誤解讀。較小的目標故障數會造成報告的故障覆蓋率虛高。這是理解故障覆蓋率的一個關鍵點，也因如此，務必要瞭解為檢測故障向量集而選擇的故障模型。

$$\text{故障覆蓋率} = \text{檢測出的故障數} \div \text{目標故障數}$$

$$\text{總故障覆蓋率} = \text{檢測出的故障數} \div \text{所有故障數}$$

例如：對於理想化元器件，“固定”引腳故障模

型可對 100 個潛在的引腳故障進行識別。而對於同一元器件，“固定”邏輯模型可對 1000 個潛在的固定故障進行識別。對這兩種故障模型分別進行類比後可知，引腳故障模型的覆蓋率為 95%，而邏輯固定模型的故障覆蓋率為 90%。哪種模型的故障覆蓋率更高、測試品質更佳？

我們來評估一下。引腳故障模型的覆蓋率達到 95%，即，總共檢測出 95 個故障 (95/100)。固定故障模型的覆蓋率達到 90%，即，總共檢測出 900 個故障 (900/1000)。顯然，固定邏輯故障方法檢測出的故障更多，但故障覆蓋率卻比較小。如果理論上可測故障總共為 1000 個邏輯固定故障，則引腳模型的總故障覆蓋率僅為 9.5% (總共 1000 個故障，檢測出 95 個故障)。在本例中，引腳故障模型最初報告的覆蓋率為 95%，而實際其總覆蓋率比報告的 90% 覆蓋率低了將近一個數量級。這是一種錯誤解讀故障覆蓋率的方式，也因如此，瞭解所選目標模

型就顯得非常重要。

## “什麼是殘餘缺陷？”

殘餘缺陷是指測試過程完成後，通過測試的合格器件中可能遺留的潛在缺陷狀況。此類缺陷並非測試的“漏網之魚”，而是由於潛在缺陷的故障目標不存在而未經測試的潛在故障狀況。殘餘缺陷和殘餘缺陷率在統計學上具有重要意義，因為它們可以預測元器件失效、PCB 板失效以及系統級失效的概率。被測物料批次的殘餘缺陷率可根據故障覆蓋率和被測產品良率而確定。

殘餘缺陷率 (以 DPM 為單位) =  $(1 - Y^{(1-T)}) \bullet 10^6$

其中：T 是小數形式的總故障缺陷率

Y 是小數形式的測試良率

重要提示：此處小數形式的故障覆蓋率必須是檢測出的故障數與理論上可測故障總數的比值，而不是相對於所選故障模型的比值。

表：分級的覆蓋率及總覆蓋率

分級的覆蓋率	
引腳固定模型：	檢測出 95 個故障 ÷ 100 個目標故障 = 95% 的故障覆蓋率
邏輯固定模型：	檢測出 900 個故障 ÷ 1000 個目標故障 = 90% 的故障覆蓋率
真正的總覆蓋率是多少？	
引腳固定模型：	檢測出 95 個故障 ÷ 1000 個目標故障 = 9.5% 的故障覆蓋率
邏輯固定模型：	檢測出 900 個故障 ÷ 1000 個目標故障 = 90% 的故障覆蓋率

表：元器件殘餘缺陷率 (以每百萬缺陷數表示)。

下期待續…… CTA

## 元器件殘餘缺陷率 (以 DPM 表示)

測試良率												測試故障覆蓋率 (佔理論可測故障總數的百分比)											
		10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	99%	100%										
99.9%	900	800	700	600	500	400	300	200	100	50	10	0											
99.5%	4501	4002	3503	3003	2503	2003	1503	1002	501	251	50	0											
99%	9005	8008	7011	6012	5013	4012	3011	2008	1005	502	100	0											
98%	18018	16032	14042	12048	10051	8049	6042	4032	2018	1010	202	0											
97%	27041	24073	21096	18110	15114	12110	9096	6073	3041	1522	305	0											
96%	36073	32130	28171	24196	20204	16196	12172	8131	4074	2039	408	0											
95%	45115	40204	35268	30307	25321	20308	15270	10206	5116	2561	513	0											
94%	54166	48295	42388	36445	30464	24446	18391	12299	6168	3089	619	0											
93%	63226	56403	49531	42608	35635	28611	21536	14409	7231	3622	725	0											
92%	72297	64529	56696	48798	40834	32803	24704	16538	8303	4160	833	0											
91%	81377	72673	63885	55015	46061	37022	27897	18685	9387	4704	943	0											
90%	90467	80834	71098	61260	51317	41268	31114	20852	10481	5254	1053	0											