



針對用戶可規劃閘陣列的面積與速度最佳化暨分割之研究 A Study of Area/Performance Optimization and Partitioning for Field Programmable Gate Array

計劃編號: NSC87-2215-E-009-040

執行期限: 86/8/01-87/07/31

主持人: 沈文仁教授/交通大學電子研究所

摘要

Keyword :

Field Programmable Gate Array (FPGA), FPGA Synthesis, FPGA Partitioning, Circuit Clustering, Area/Performance Trade-Off, Constraint-Driven Synthesis

● 中文摘要

本計畫中，我們以邏輯電路實作於用戶可規劃閘陣列之設計為主題，分成三個子題來進行研究。第一個子題是建立一套可在用戶可規劃閘陣列上提供從面積最佳化至速度最佳化之映射解集合的合成系統，以供設計者從中選擇一個符合其面積、速度要求之最佳解。由於傳統的用戶可規劃閘陣列之合成技術僅單獨考慮面積最佳化，或者是速度最佳化，故無法提供以限制驅動為基礎之合成設計。在本子題中我們成功地發展了一套新的用戶可規劃閘陣列之合成系統，提供所有在面積/速度取捨區線上的設計解，供設計者在預定的規格下選擇最佳解。第二個子題是設計一套電路叢集系統，我們利用一種多向無回路電路叢集技術，以不同的叢集數當作參數來得到一組在容量限制下的叢集數/速度取捨之解集合。第三個子題是發展用戶可規劃閘陣列之分割系統；同時考慮用戶可規劃閘陣列的邏輯容量限制和接腳數的限制，並且利用前兩個子題之成果，發展出可進行面積/速度取捨之用戶可規劃閘陣列之分割系統。後兩個子題我們已有相當可觀的進展。

● Abstract

In this project, we studied three topics about how to implement a logic circuit by using FPGAs. The first topic is to establish an FPGA synthesis system to provide a set of solutions from area-optimized one to performance-optimized one. Then, the designer can choose the best solution from them to meet the area and performance constraints. Because the traditional FPGA synthesis techniques consider either the area optimization or the performance optimization only, they cannot provide the constraint-driven based synthesis style. In this topic, we have successfully developed a new FPGA synthesis system which can generate a set of comprehensive design solution for the designer under predefined specifications. The second topic is to design a circuit clustering system. We take advantage of an acyclic k-way clustering technique. Iteratively use different k's as parameters, a set of #cluster/performance trade-off solutions under the capacity constraint can be obtained. The third topic is to develop an FPGA partitioning system. We take both the logic capacity and pin constraints into consideration. Adopting the techniques developed in the first and second topic, we will develop an FPGA partitioning system which satisfies both the capacity and pin constraints as well as provides the area/performance trade-off partitioning solutions. We have had considerable

progress in the latter two topics.

二. 計畫緣由與目的

面對半導體製程技術的快速發展，積體電路的密度大幅度提升，使得一顆晶片內可含的邏輯閘數目快速增加，於是設計者可將更複雜的系統整合入一顆單晶片。但是為了要確保晶片設計的正確性，在目前的設計流程下，除了要作軟體模擬(simulation)之外，還要作硬體模擬(emulation)。而用戶可規劃陣陣列(Field Programmable Gate Array, FPGA)這種新的 VLSI 元件，因為它的高邏輯密度、可程式、可擴充性等優點，使得它極適合運用在建構硬體模擬系統(hardware emulation system)及快速的產品雛形(Prototype)製作。

傳統上，要將一 VLSI 電路植入硬體模擬系統，其流程為第一步先執行為一般 ASIC 電路而設計的與技術映射無關(technology-independent)最佳化，再將所得的結果針對實現於 FPGA 做特別的技術映射(technology mapping)，在做完技術映射後，所得的結果若無法用單獨一顆 FPGA 來實現，則需要將所得的映射結果分割成幾個子線路，而任何一個子線路必須可被一顆 FPGA 晶片單獨實現。

在這個流程下，將有三個問題需要解決，第一點是如何為不同於傳統 ASIC 電路的 FPGA 做技術映射。第二點是如何有效地解決 FPGA 之分割問題。第三點是如何在 FPGA 的技術映射與分割之間做完美緊密的結合，我們將繼續針對這三個問題，做更深一層的探討：

(一)FPGA 技術映射問題：

在傳統的 ASIC 電路設計中，做完 technology-independent 最佳化步驟之後，緊接的是技術映射最佳化，針對查對表式 FPGA，已有一些技術映射的演算法被提出，

但他們不是僅針對面積做最佳化，就是只對速度做最佳化，所以並無法提供限制驅動式(constraint-driven)的映射結果。

所以發展一套可求得輸入電路設計所有位於面積/速度取捨曲線(trade-off curve)上的映射解，使得原設計者可在預定的面積和速度限制下挑出一個最佳解的系統是非常重要的而且實用的。這套軟體可幫助設計者輕易建構一個符合自己特殊需求的硬體模擬系統，只要輸入原始設計及面積或速度的限制，則這套軟體將會直接傳回符合設計條件的最佳映射解，進而建構出最佳之硬體模擬系統。

(二)FPGA 分割的問題：

因為目前的 FPGA 元件，有一定的邏輯容量及接腳數的限制，所以當一個設計者無法用單獨一顆 FPGA 實作時，則輸入電路必須分割(partition)成數個子電路，使得每一子電路皆可被一顆 FPGA 所實作。在 FPGA 分割的問題中最需要解決的是速度瓶頸問題，因為當信號跨越不同的 FPGA 晶片時，會引入極大的晶片間延遲(inter-chip delay)，而造成速度極大的損失，而且這個速度延遲在 FPGA 技術映射時並無法預估，只有在分割之後才能準確得知。所以如何分割電路使其尚能保持可接受之電路速度，亦是建構 FPGA 所組成之硬體模擬系統所要考慮的重要關鍵之一。目前為止，已有一種在容量限制之下，可求得輸入電路之最佳速度分割解的技術被提出來。然而經過我們對其實驗結果做進一步的分析，發現其雖可求得速度最佳化之解，但因為此法採用了大量的邏輯複製以及為了滿足最佳速度之要求，其所需要的額外面積(area overhead)相當地大，所以這個技術可能無法非常實用地解決現實上的分割問題。而且這種技術尚有一種缺點，那就是它僅能求得速度最佳之分割解，但是卻無法求得所有在分割數/速度取捨曲線上的所有分割解供設計者選擇。所以與在技術映射層面上一樣，

如果有一組從分割數最佳化至速度最佳化一應俱全之解集合提供給設計者，讓其在預定的規格下選擇一個符合限制的最佳解，應該是比較完美的解決之道。所以在本計劃中我們打算採用一種新的技術，稱為 K 向無回路式電路叢集法(K-way acyclic circuit clustering)來處理電路的分割問題。藉由代入不同的 K 當參數來取得一系列位於分割數/速度取捨曲線上的分割解。若實驗結果證明由此法所求得的最佳速度分割解相當於理論最佳值，卻又可大幅減少額外面積的需求，且又因為此法可提供限制驅動式之分割結果，所以我們所提出的這個新分割技術，可以全盤解決在 FPGA 分割上所遭遇的問題。

(三)FPGA 技術映射與分割之間的整合問題

傳統上，技術映射及電路分割在設計的流程之中是兩個獨立的步驟，之間並無任何的連繫整合。這種作法會喪失得到真正最佳化的結果，即使是在獨立的兩個子步驟中都能取得最佳解，舉例來說，若在技術映射時欲得速度最佳化之解，但在未分割之前根本無法確定電路於何處會發生晶片間的延遲，故無法提供準確的時間資訊，因此當然無法求得真正的速度最佳解。要解決這個問題的關鍵，應在於如何完美緊密地結合技術映射及分割兩個步驟，成為一個整合單獨的步驟。如此才能在技術映射時考慮分割的效應，而在完成技術映射之後自然就是分割好的結果。所以在這個計劃之中，我們嘗試把針對問題(一)和(二)所發展的技術，做一個緊密地連結，來完成一個整合的系統。它會將輸入的電路直接映射成一個由多顆 FPGA 所組成的硬體模擬系統而且是做限制驅動式的合成。

三. 研究方法與結果

第一年的計畫中，我們著重於技術映射的部份。對於這個問題，傳統的技術皆著重於

面積或速度做最佳化，於此我們發展一種新的技術能夠完整地求得從面積最佳化至速度最佳化之解集合，以完成限制驅動式的合成。我們先求得一經速度考量的面積最佳解(Performance-considered area-optimized solution)，再對此結果做時間分析(timing analysis)找出電路的臨界區域(critical section)，再對此區域施以速度最佳化的重合成技術(performance optimization resynthesis techniques)，如此一再地反覆(iteratively)執行上述步驟，便可將一面積最佳化之映射解，逐漸地推至速度最佳化解，而取得完整的面積/速度取捨曲線，以完成限制驅動式的合成。雖然要取得完整的面積/速度取捨曲線，理論上亦可經由速度最佳化之解出發，經過一系列以速度交換面積的步驟來完成。但是我們發現在 ASIC 電路設計上，現存的設計流程及技術皆不採取此種作法，故此種方法必有其窒礙難行之處。所以我們採取與傳統 ASIC 電路設計類似的流程，以吸取其已發展之經驗及技術，提高成功的可能性。

在求取 performance-considered area-optimized solution 方面，我們採取將時間資訊整合入面積最佳化的技術中，使得所求得之解不僅是面積最佳化，且在速度上亦有不錯的表現。在 performance optimization resynthesis techniques 方面，我們則改良一些現有的技術並發展一些新的技術，來以最少額外面積換取速度。當完成這兩大項技術研發後，就可以達成我們建立限制驅動式 FPGA 技術映射系統的目標。

整個技術映射的系統都已經架構在 SIS 的環境中。表(一)的結果顯示我們的演算法能夠有效地提供完整的面積/速度取捨曲線。有時候最佳的面積和速度會同時出現，因此也就沒有取捨曲線，如 5xpl, 9sym, 和 9symml 這三個電路。而效果最顯著的則是 e64, 電路層次從 17 一直減少到 3, 圖(一)

是它的取捨曲線。表(二)是我們的演算法(ALTO)和另一個可作面積/速度取捨曲線的演算法(FlowMap-r)的比較,在大部分的電路中,在相同的速度下,ALTO 都有較少的面積。最後,ALTO 在和純粹作速度最佳化的演算法比較時也有很好的表現,如表(三)。

表(二): ALTO 和 FlowMap-r 的比較

CKT	L		L-1		L-2		L-3		L-4		
	#LVL	ALTO	Fmap-r	ALTO	Fmap-r	ALTO	Fmap-r	ALTO	Fmap-r	ALTO	Fmap-r
5xpl	4	-	22	-	23	19	-	-	-	-	-
C499	7	70	-	70	130	78	151	-	-	-	-
C880	11	-	172	93	179	93	195	96	211	96	-
alu2	9	-	140	-	148	-	-	61	-	66	-
alu4	12	181	240	199	244	208	245	209	-	259	-
apex6	8	209	-	199	-	201	220	208	221	211	232
apex7	6	62	-	62	76	66	80	81	-	-	-
count	5	31	57	32	73	43	-	-	-	-	-
des	9	858	934	845	969	801	987	784	1003	-	1087
duke2	7	116	151	117	161	128	172	156	187	-	-
rd84	6	-	38	-	42	-	43	13	-	-	-
rot	10	193	-	199	210	204	213	214	218	-	243

四. 結論與討論

在今年的計畫中,我們完成了一套可求得面積/速度取捨曲線的用戶可規劃陣列技術映射演算法 ALTO。我們由一個經速度考量的面積最佳解出發,反覆地對電路的臨界區域施以速度最佳化的重合成技術,而得到完整的面積/速度取捨曲線。除此之外,ALTO 對速度的最佳化也有非常好的效果,由於ALTO 原本是針對面積/速度取捨曲線來設計,能夠在速度上贏過絕大部份專門針對速度作最佳化考慮的演算法,是計畫意外的收穫。

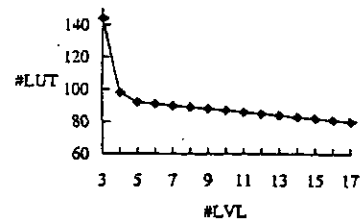
表(三): ALTO 和其他速度最佳化的演算法比較

CKT	charte-d		misga-d		TechMap		FlowMap		FlowSYN		DOGMA		BoolMap		ALTO	
	#LUT	#LVL	#LUT	#LVL	#LUT	#LVL	#LUT	#LVL	#LUT	#LVL	#LUT	#LVL	#LUT	#LVL	#LUT	#LVL
5xpl	26	3	21	2	17	2	22	3	20	2	24	3	13	2	19	2
9sym	63	5	7	3	9	3	60	5	7	3	49	5	7	3	7	3
9symml	59	5	7	3	9	3	55	5	7	3	50	4	7	3	7	3
C499	382	6	199	8	148	4	68	4	133	5	68	4	101	4	78	5
C880	329	8	259	9	213	7	124	8	232	8	98	8	146	7	96	7
alu2	227	9	122	6	197	8	155	9	113	6	138	9	43	4	68	5
alu4	500	10	155	11	-	-	253	9	249	9	-	-	268	7	259	8
apex6	308	4	274	5	252	5	238	5	257	4	231	5	189	4	211	4
apex7	108	4	95	4	86	4	79	4	89	4	68	4	78	3	81	3
b9	-	-	47	3	49	4	-	-	-	-	-	-	-	-	36	3
clip	-	-	54	4	40	3	-	-	-	-	-	-	-	-	33	3
count	91	4	81	4	71	4	31	5	75	3	31	5	42	3	43	3
des	2086	6	1397	11	1395	8	1310	5	893	4	938	5	594	3	784	6
duke2	241	4	164	6	175	4	174	4	187	4	173	4	193	5	156	4
e64	139	7	212	5	218	4	-	-	-	-	-	-	-	-	144	3
f51m	-	-	23	4	38	4	-	-	-	-	-	-	-	-	15	3
misex1	19	2	17	2	18	2	16	2	15	2	16	2	15	2	14	2
misex2	-	-	37	3	41	3	-	-	-	-	-	-	-	-	37	2
misex3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	251	6
rd73	-	-	8	2	10	2	-	-	-	-	-	-	-	-	8	2
rd84	61	4	13	3	16	3	46	4	13	3	53	4	10	2	13	3
rot	326	6	322	7	315	6	234	7	262	6	210	7	228	6	214	7
sao2	-	-	45	5	45	4	-	-	-	-	-	-	-	-	33	3
vg2	55	4	39	4	36	4	29	3	45	4	27	3	30	4	26	3
z4ml	25	3	10	2	9	2	5	2	6	2	5	2	5	2	5	2
charte-d	5043	94	-	-	-	-	-	-	-	-	-	-	-	-	2225	73
misga-d	-	-	3608	116	-	-	-	-	-	-	-	-	-	-	2387	89
TechMap	-	-	-	-	3407	91	-	-	-	-	-	-	-	-	2128	81
FlowMap	-	-	-	-	-	-	2899	84	-	-	-	-	-	-	2081	70
FlowSYN	-	-	-	-	-	-	-	-	2601	72	-	-	-	-	2081	70
DOGMA	-	-	-	-	-	-	-	-	-	-	2189	74	-	-	1822	62
BoolMap	-	-	-	-	-	-	-	-	-	-	-	-	1969	63	2081	70

五. 圖表

表(一): 面積/速度取捨曲線實驗結果

CKT	L _{opt}	L	L-1	L-2	L-3	L-4	L-5	L-6	L-7	L-8
	#LVL	#LUT	#LUT	#LUT	#LUT	#LUT	#LUT	#LUT	#LUT	#LUT
5xpl	2	19	-	-	-	-	-	-	-	-
9sym	3	7	-	-	-	-	-	-	-	-
9symml	3	7	-	-	-	-	-	-	-	-
C499	7	70	70	78	-	-	-	-	-	-
C880	10	93	93	96	96	-	-	-	-	-
alu2	6	61	66	-	-	-	-	-	-	-
alu4	14	162	180	181	199	208	209	259	-	-
apex6	8	209	199	201	208	211	-	-	-	-
apex7	6	62	62	66	81	-	-	-	-	-
b9	4	35	36	-	-	-	-	-	-	-
clip	5	31	32	33	-	-	-	-	-	-
count	5	31	32	43	-	-	-	-	-	-
des	13	849	852	859	876	858	845	801	784	-
duke2	7	116	117	128	156	-	-	-	-	-
e64	17	80	81	82	83	84	85	86	87	88
f51m	3	15	-	-	-	-	-	-	-	-
misex1	3	15	14	-	-	-	-	-	-	-
misex2	3	32	37	-	-	-	-	-	-	-
misex3	13	144	149	161	167	173	201	218	251	-
rd73	2	8	-	-	-	-	-	-	-	-
rd84	3	13	-	-	-	-	-	-	-	-
rot	12	187	188	193	199	204	214	-	-	-
sao2	5	37	34	38	-	-	-	-	-	-
vg2	4	22	26	-	-	-	-	-	-	-
z4ml	2	5	-	-	-	-	-	-	-	-



#LVL	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3
#LUT	80	81	82	83	84	85	86	87	88	89	90	91	92	98	144

圖(一): e64 電路的面積/速度取捨曲線