

共用匯流排系統之高階設計

High-Level Synthesis of Shared-Buses Systems

計畫編號：NSC 88-2213-E-009-009

執行期限：87年8月1日至88年7月31日

主持人：蔡中川 交通大學資訊工程系

一、中文摘要

共用匯流排系統為多個功能單元共用多個匯流排之數位系統，給定一個代表此系統行為之資料流動圖，其高階設計工作至少包含時間排程與資源安排，本文中將提出這些問題之最佳解方法。我們的方法除可應用於先前單位計算與資料傳輸時間之非重複式資料流動圖外，還可應用於非單位計算與資料傳輸時間之重複式資料流動圖。對於排程問題，我們提出擴充式資料流動圖之觀念，使得資料傳輸可視同如計算，而由單一整數規劃程式解決其排程問題；另外，我們還利用整數規劃、圓弧著色和通道指派技術解資源安排問題。

關鍵詞：資料流動圖，高階設計，整數規劃，最小成本網路流，資源安排，共用匯流排系統，時間排程

Abstract

A shared-bus system consists of multiple buses shared by multiple functional units. Two important problems involved in the high-level synthesis of such a system are: the scheduling problem and the resource allocation problem for a given data-flow graph (DFG). This report proposes methods to solve these two problems optimally. The proposed methods are applicable to both non-iterative DFGs and iterative DFGs with non-unitary communication and computation times. To solve the scheduling problem, we introduce the extended DFG so that both computations and communications can be scheduled simultaneously to functional units and interconnection buses in a single integer linear programming formulation. To solve the resource allocation problem, we apply techniques of integer linear programming circular-arc coloring and channel assignment.

Keywords: data-flow graphs, high-level synthesis, integer linear programming, minimum-cost network flow, resource allocation, shared-bus system, time

scheduling

二、緣由與目的

隨著超大型積體電路技術之進步，特定應用之電路設計愈來愈複雜，直接進行低階電路設計很容易出錯且繁複，所以高階設計(high-level synthesis)成為重要之程序。高階設計在過去十數年間，是一個很活躍之研究領域[1,2]，其中時間排程(time scheduling)和資源安排(resource allocation)是高階設計之兩項主要工作。時間排程探討如何在指定之資源(或完成時間)要求下，以最短完成時間(或最少資源)將運算指派至特定時間執行——稱為資源限制(時間限制)排程。資源安排則探討如何將指定數量之資源安排給已排程之運算。

一般而言，高階設計問題都是很困難的問題。在有足夠資源之前提下，利用展開(unfolding)和重新定時(retiming)技術，最佳速率排程有多項式計算複雜度之演算法[3,4]；有些研究探討以計算時間短之啟發式解法得到近似最佳時間排程和資源安排[5,6,7,8,9]；有些研究則利用計算複雜度較高之整數規劃方法找尋最佳解，實證發現整數規劃方法仍有很高之應用價值[10,11,12]，因此許多研究[13,14,15,16]也利用這種方法解答特定架構之最佳設計問題。雖然這些最佳化設計方法在高階設計特定架構上得到很好成果，但[10,11,12]侷限於非重複式演算法之探討，[13,14,16]不考慮資料傳遞時間；雖然[15]在非重複式演算法共用匯流排架構高階設計時將資料傳遞時間列入考慮，但他們須列舉各種資源組合，並對每一組合解一個整數規劃問題，除了須列舉大量資源組合(解許多整數規劃問題)外，常會得不到最佳解。

綜合言之，過去的研究或專注於非重複式之演算法和單位計算時間假設、或非最佳解之啟發式解法、或忽略資料傳遞時間、或計算方法可進一步改

進。本研究將包含重複式演算法與非重複式之演算法、非單位運算時間及非單位資料傳遞時間等假設，提出其資源限制與時間限制排程之設計方法；並利用展開與最小成本網路流之技術，設計更有效之資源安排計算方法。

三、結果與討論

本研究中我們將一個演算法之共用匯流排高階設計表示成，由代表演算法之資料流動圖 (data-flow graph, DFG) $G=(V,E,d,c,b)$ 得到一個共用匯流排高階設計 $S=(\tau,R)$ 。資料流動圖 G 中之 V 是一個節點集合； E 是一個有方向性鏈路集合；對於每一個節點 $v \in V$ ， $c(v)$ 代表節點 v 之整數執行時間；節點 v 之第 i 批次(iteration) 表示成 v^i ，我們稱之為一個運算，對於每一個鏈路 $e \in E$ ； $d(e)$ 代表鏈路 e 之批次差；一個批次差為 $d(e)=j-i$ 之鏈路 $e=(u,v)$ 代表著，從運算 u 至運算 v^i 有一個執行先後順序之限制； $b(e)$ 代表鏈路 e 所需之資料傳遞時間。設計 S 中之 T 是它的時間排程， R 是它的資源安排， T_v^i 和 R_v^i 分別代表運算 v^i 之開始執行時間及執行該運算之資源。

每個節點都僅執行一次的 DFG 被稱為非重複式 DFG (non-iterative DFG, NDFG)，否則稱為重複式 DFG (iterative DFG, IDFG)，執行一個批次運算之平均時間稱為批次週期(iteration period)，整個時間排程(運算及資料傳遞)所涵蓋之時間區間稱為延滯時間(latency)。給定批次週期後，節點 v 之最早及最遲執行時間分別表示為 s^v 及 l^v ，計算一個 DFG 批次週期之複雜度為 $O(|E||V|)$ ，而計算 NDFG 之延滯時間則僅為線性複雜度 [6]。一個滿足 $T_v^i = T_v^{i-r} + p$ 和 $R_v^i = R_v^{i-r}$ 條件之設計，是一個具有展開度(unfolding factor) r ，時間位移 p 之全靜態設計(fully-static design)，其批次週期為 $\frac{p}{r}$ ，那麼完整設計可以重複連續 r 批次之設計得到。

3.1 擴充式 DFG

在共用匯流排架構下，DFG 中每一節點(運算)由指定種類之功能單元執行，DFG 中鏈路(運算與運算間)之資料傳遞由共用匯流排執行，因此共用匯流排之高階設計必需同時考慮節點和鏈路。為能簡化設計作業，我們將資料傳遞當成使用匯流排(視之為一種運算單元)之運算，將鏈路轉換成節點後之

DFG 稱為擴充式 DFG (Extended DFG)，因此設計工作不需同時處理 DFG 中節點及鏈路，僅需處理擴充式 DFG 之節點即可，因此除非特別指明，否則以下所討論之 DFG 皆為擴充式 DFG。

給定一 DFG $G=(V,E,d,c,b)$ ，建立其對應擴充式 DFG $G^e=(V^e,E^e,d^e,c^e)$ 的方法如下：(1) 對於每一鏈路 $e=(u,v) \in E$ ，我們新增一節點 w ，並以鏈路 (u,w) 和 (w,v) 取代原鏈路 e ；(2) 設 $d^e(u,w)=0$ ， $d^e(w,v)=d(u,v)$ ；(3) 設 $c^e(u)=c(u)$ 且 $c^e(w)=b(u,v)$ 。另外，我們定義 $F=\{F_1,F_2,\dots\}$ 為一組資源種類之集合，資源種類 F_i 可能是一種功能單元或匯流排，節點 $v^e \in V^e$ 所須之資源則表示成 $f(v^e)$ 。

圖 1(a) 是一個 13 個輸入 2 個輸出之 NDFG [15]，節點 $v_1, v_2, v_3, v_4, v_{11}, v_{12}$ 使用資源種類 F_1 (乘法器)，節點 v_9 使用資源種類 F_2 (除法器)，其它節點則使用資源種類 F_3 (加法器)，而圖 1(b) 為圖 1(a) 之擴充式 DFG。圖 2(a) 是一個 4 個輸入 1 個輸出之 IDFG [17]，鏈路 e 上的短槓數代表 $d(e)$ ，節點 $v_1, v_2, v_5, v_6, v_8, v_{11}$ 使用資源種類 F_1 (乘法器)，節點 $v_3, v_4, v_7, v_9, v_{10}, v_{12}$ 使用資源種類 F_2 (加法器)，而圖 2(b) 為圖 2(a) 之擴充式 DFG。兩擴充式 DFG 中每一新加節點 w_i 對應一個資料傳遞，本文中我們假設乘法器與除法器需花 2 個單位時間，而加法器及資料傳遞僅需花 1 個單位時間。

3.2 NDFG 之時間排程

給一個 NDFG G 和延滯時間 T ，我們可以得到每一個節點 $v \in V$ 之排程範圍 $[s^v, l^v]$ ，在不損失其一般性下，我們假設 $\min_{v \in V} s^v = 0$ 和 $\max_{v \in V} (l^v + c(v)) = T$ 。當 $j \in [t, t+r-1]$ 時我們設 $m(j, t, r) = 1$ ；否則 $m(j, t, r) = 0$ 。如此，時間限制之排程問題可表示成式子 (1)-(5) 之整數規劃模式。其中 N_δ 代表資源 δ 之個數； W_δ 為加權數。

$$\text{以圖 1(a) 為例，最小延滯時間為 } 14_\delta, \text{ 以 } T=14 \text{ 解式子 (2)-(5) 之時間限制排程之整數規劃問題，得到最少須要兩個乘法器、一個除法器、一個加法器和四條匯流排，排程結果見圖 3(a), 3(b)。} \quad (6)$$

$$\text{NDFG 資源限制之排程問題亦可表示成類似之整數規劃模式} \quad (7)$$

$$\text{IDFG 之時間排程} \quad (8)$$

$$\sum_{t=s^u}^{l^u} \sum_{t=s^v}^{l^v} m(j, t, r) * x_{ut} \leq N_\delta \quad (9)$$

$$\sum_{t=s^u}^{l^u} \sum_{t=s^v}^{l^v} m(j, t, r) * x_{ut} \leq N_\delta \quad (10)$$

設 h, T, t, s 為整數且 $h \in [0, T-1]$ ，我們定義 $n(T, h, t, s)$ 為在 $j \in [t, t+s-1]$ 範圍內且 $j \bmod T \equiv h$ 之整數 j 之個數；那麼 $n(T, h, t, s) = \lfloor \frac{s}{T} \rfloor + n(T, h, t, s - \lfloor \frac{s}{T} \rfloor * T)$ 。若 T 是指定之批次週期，則 IDFG 之時間限制之排程問題可表示成式子(6)-(10)之整數規劃模式。

以圖 2(a)為例，利用[18]中演算法可得到它的最佳批次週期為 5，取 $T=5$ 解式子(6)-(10)之時間限制整數規劃問題，得到排程如圖 4(b)-4(e)。

3.4 NDFG 之資源安排

NDFG 之排程包含每一節點之開始執行時間 $\{T_v | v \in V\}$ 和資源集合 $\{(N_\delta, \delta) | \delta \in F\}$ 。資源安排將已排程之節點安排至特定的資源上，這種安排可表示成 $\{R_v | v \in V\}$ 。每一運算在特定時間區間被特定之資源執行，被指派至同一個資源之兩運算，它們的執行區間不可以重疊。最少資源安排問題在探討如何以最少資源執行所有已排程之運算，這個問題等於 Gupta et al [19]所稱之通道指派問題(channel assignment problem)，通道指派方法可以得到最佳解，其計算複雜度為 $O(N \log N)$ ，其中 N 為節點數。

以圖 1(a)之 NDFG 為例，選延滯時間 $T=14$ ，首先解式子(2)-(5)得到時間限制排程：再去解每一種資源之資源安排，結果如圖 3(a)-(d)。圖 3(e)是它的共用匯流排設計結果。

3.5 IDFG 之資源安排

因為 IDFG 中節點包含一個以上批次之運算，其高階設計可表示成一個週期式排程 $T_v^i = T_v^{i-1} + l_p$ (批次週期為 l_p) 和一個週期式資源安排 $R_v^i = R_v^{i-r}$ (r 是展開度)。結合週期式排程及週期式資源安排，我們得到一個全靜態設計 $S = \{T, R\}$ ，其時間排程為 $T_v^i = T_v^{i-r} + r * l_p$ ，資源安排為 $R_v^i = R_v^{i-r}$ 。因此，我們可以重複連續 r 個批次之排程及資源分派就可得到完整之設計 S ；換句話說，我們只要設計連續 r 個批次即可。為簡化問題，我們僅考慮在區間 $[0, r * l_p - 1]$ 內開始被執行之 r 個批次。接著我們可將其轉成圓弧著色(circular-arc coloring)問題[8]。不過[此種問題屬於難解的 NP-complete 問題[20,21]。

另一種資源安排方法為採用整數規劃技巧。給一排程 $\{T_v | v \in V\}$ 和 N_δ 個資源種類 $\delta \in F$ 。以 $r=1$ 為例，資源安排問題可表示成式子(11)-(13)之整數規劃模式。

圓弧著色及整數規劃兩種方法都可得到最小展開度之設計，但若放鬆最小展開度之要求，我們可利用通道指派演算法或最小成本流量演算法解資源安排問題[22,23]。按照批次週期區間 $[0, l_p - 1]$ ，將於該區間開始執行且執行區間跨越該區間之節點，依照批次週期區間分割成數個區間，再將分割之區間重新結合起來。以圖 4(a)之時間排程為例 ($l_p = 4$)，將被分割節點重新結合後，我們可找到具週期性之資源安排如圖 4(b)-(e)，圖 4(f)是它的共用匯流排設計結果。

四、計劃成果自評

本計劃成果已達預期目標。研究成果發表於 *Proceedings of NSC, Part A* (vol 23, 1999)。

五、參考文獻

- [1] R. A. Walker and R. Camposano, *A Survey of High-Level Synthesis Systems*, Kluwer Academic Publishers, Norwell, MA, U.S.A., 1991.
- [2] D. D. Gajski, N. D. Dutt, A. C. H. Wu, and S. Y. L. Lin, *High-Level Synthesis: Introduction to Chip and System Design*, Kluwer Academic Publishers, Norwell, MA, U.S.A., 1992.
- [3] K. K. Parhi and D. G. Messerschmitt, "Static rate-optimal scheduling of iterative data-flow programs via optimum unfolding" *IEEE Trans. Comput.*, vol 40, no. 2, pp. 178-195, 1991
- [4] L. F. Chao and Edwin H.M. Sha, "Static scheduling for synthesis of DSP algorithms on various models," *Journal of VLSI Signal Processing*, vol 10, pp. 207-223, 1995.
- [5] P. G. Paulin and J. P. Knight, "Force-directed scheduling for the behavioral synthesis of ASIC's," *IEEE Trans. Computer-Aided Design*, vol 8, no. 6, pp. 661-679, 1989.
- [6] S. M. H. H. Groot and O. E. Hermann, "Range-chart-guided iterative data-flow graph scheduling" *IEEE Trans. Cir. and Syst.-I*, vol 39, pp. 351-364, May 1992.
- [7] L. G. Jeng and L. G. Chen, "Rate-optimal DSP synthesis by pipeline and minimum unfolding" *IEEE Trans. VLSI Systems*, vol 2, no. 1, pp. 81-88,

$$\sum_{w=1}^{N_S} \gamma_{vw} = 1$$

$$\{v \in \mathcal{V}, f(v) = \delta\} \quad (11)$$

$$\gamma_{uv} + \gamma_{vw} = 1$$

$$\{u, v | T_u + c(u) - T_v > l_p; f(u) = f(v) = \delta, u, v \in \mathcal{V}\};$$

$$w = 1, \dots, N_S \quad (12)$$

$$\gamma_{vw} = 0 \text{ or } 1 \quad (13)$$

1994.

[8] D. J. Wang and Y. H. Hu, "Multiprocessor implementation of real-time DSP algorithms," *IEEE Trans. VLSI Systems*, vol. 3, no. 3, pp. 393-403, 1995.

[9] C. Y. Wang and K. K. Parhi, "Resource-constrained loop list schedule for DSP algorithms," *Journal of VLSI Signal Processing* vol. 11, pp.75-96, 1995.

[10] J. H. Lee, Y. C. Hsu, and Y. L. Lin, "A new integer linear programming formulation for the scheduling problem in data path synthesis," in *Proceedings of the International Conference on Computer Aided Design*, Santa Clara, CA, U.S.A., pp. 20-23, 1989.

[11] C. T. Hwang, J. H. Lee, and Y. C. Hsu, "A formal approach to the scheduling problem in high level synthesis," *IEEE Trans. Computer-Aided Design*, vol. 10, no. 4, pp. 464-475, 1991.

[12] C. H. Gebotys and M. I. Elmasry, "Optimal synthesis of high-performance architectures," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 3, pp. 814-826, 1992.

[13] L. E. Lucke and K. K. Parhi, "Generalized ILP scheduling and allocation for high-level DSP synthesis," in *Proceedings of the IEEE 1993 Custom Integrated Circuits Conference*, San Diego, CA, U.S.A., pp. 541-544, 1993.

[14] K. Ito, L. E. Lucke, and K. K. Parhi, "Module selection and data format conversion for cost-optimal DSP synthesis," in *1994 IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, U.S.A., pp. 322-329, 1994.

[15] Y. Sawano, B. Kim, and M. Kameyama, "High-level synthesis of VLSI processors for intelligent integrated systems," *IEICE Trans. Electron.*, vol. E77-C7, pp. 1101-1107, 1994.

[16] R. Govindarajan, E. R. Altman, and G. R. Gao, "A framework for resource-constrained rate-optimal software pipelining" *IEEE Trans. Parallel and Distri. Sys.*, vol. 7, no. 11, pp. 1133-1148, 1996.

[17] F. B. Maciel, Y. Miyahara, and K. Tochinai, "An optimization technique for lowering the iteration bound of DSP programs," *Journal of VLSI Signal Processing* vol. 14, pp. 161-170, 1993.

[18] S. H. G. Gerez, S. M. H. H. Groot, and O. E.

Herrmann, "A polynomial-time algorithm for the computation of the iteration-period bound in recursive data-flow graphs," *IEEE Trans. Cir. and Syst.-I*, vol. 39, pp. 49-52, 1992.

[19] U. I. Gupta, D. T. Lee, and J. Y. T. Leung "An optimal solution for the channel-assignment problem," *IEEE Trans. Computer*, vol. c-28, no.

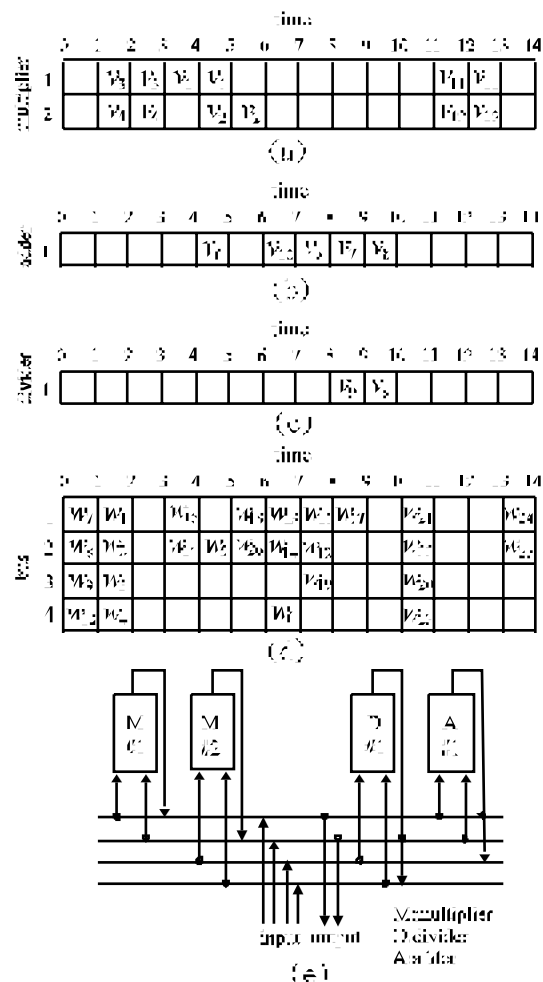


圖 5. 圖 1 的時間排程、資源安排與匯流排設計

11, pp. 807-810, 1979.

[20] A. Tucker, "Coloring a family of circular arcs," *SIAM J. Appl. Math.*, vol. 29, no. 3, pp. 493-502, 1975.

[21] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the theory of NP-completeness*, W. H. Freeman, New York, NY, U.S.A., 1979.

[22] Y. C. Ho, and J. C. Tsay, "Fully-static processor-optimal scheduling of data-flow graphs," *IEEE Signal Processing Letters*, vol. 4, no. 5, pp. 146-148, 1997.

[23] J.C. Tsay and Y.C. Ho "A network flow approach to solving weighted processor-optimal allocation problems," *Proceedings of NSC, ROC, Part A: Physical Science and Engineering*, vol. 23, no. 3, 1999.

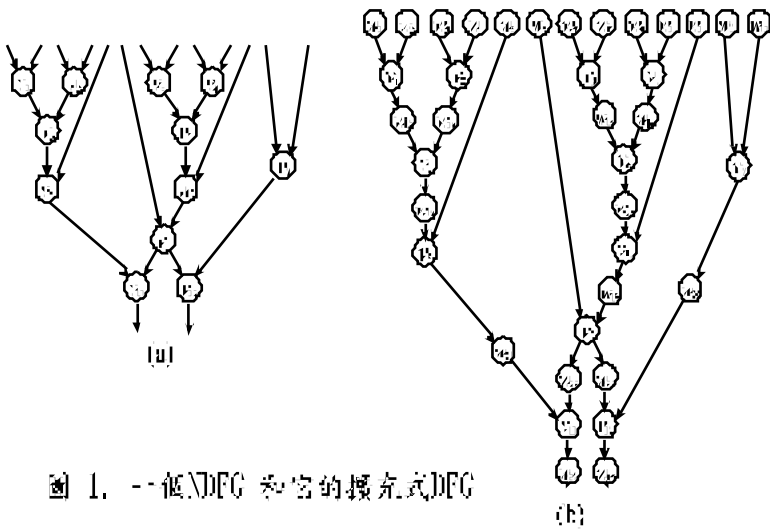


圖 1. 一個 DAG 和它的擴充式 DAG

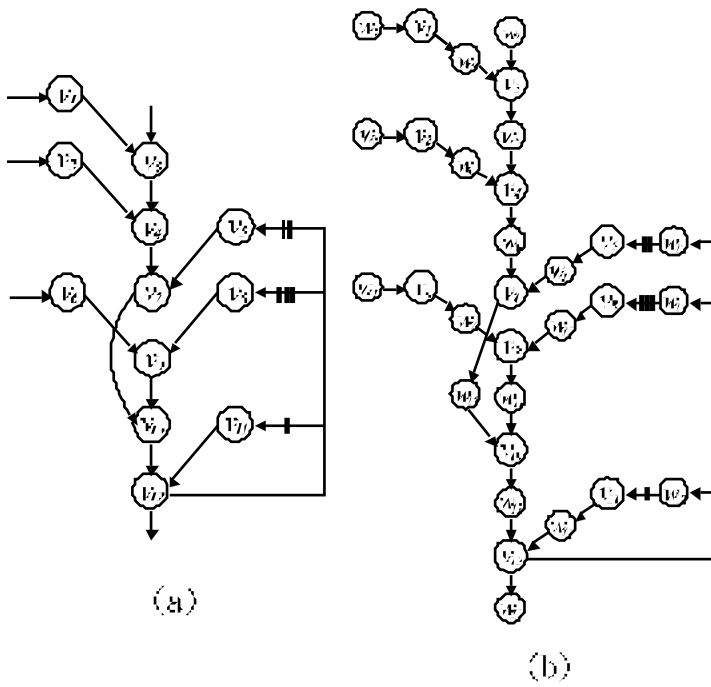


圖 2. 一個 DAG 和它的擴充式 DAG

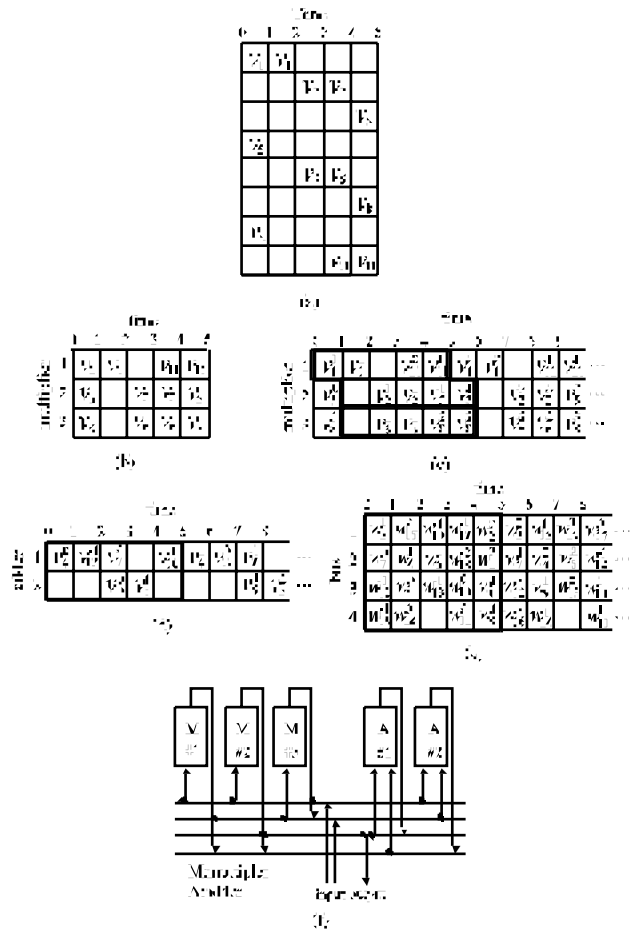


圖 4. 圖 2 的時域排程，假設每個乘器與加器執行時間