多維波數位濾波網路上的
IBM Cell 寬頻引擎之性能分析

曾建勳

崑山科技大學資訊工程系助理教授

摘要

作為物理系統數值積分的替代方法，MDWDF 網路的技術逐漸變為非常重要，這是由於它的優秀特質例如大型平行性和本身俱有的高精確度。在這份論文裡，我們針對一個在流體力學上非常重要的三維度已線性化數位濾波模型之偏微分方程式淺水波系統，來研究其達到充分並行運算後的加速效率。在 MDWDF 網路模型之充分平行運算過程中，我們使用連鎖管道技術來執行索尼(SONY) Play station 3 之 IBM Cell 寬頻引擎(Cell/BE)，並藉此提升它的運作性能。這 IBM Cell/BE 包含 1 個俱有高效率的數據處理之能量處理器元素(PPE)來提供優化的系统作用和 8 個協同作用處理器元素(SPEs)，這些元件非常適合 MDWDF 網路模型結構之時間的疊代運算，亦即要求至多 1 個主要處理器和 7 個獨立運作的加工處理器。根據計算效能的實驗性結果顯示，俱有 8 個處理器(1PPE+7SPEs)的並行化模型，其性能表現超越非平行化模組(即僅使用 PPE 的模型)有 4 倍運作速度。

關鍵詞：MDWDF、物理系统、淺水波系統、偏微分方程式、充分並行運算、IBM Cell/BE

Performance Analysis of Multidimensional Wave Digital Filtering Network on IBM Cell Broadband Engine

Chien Hsun Tseng

Department of Information Engineering, Kun Shan University of Technology, Taiwan

Abstract

As an alternative approach for the numerical integration of physical systems, the MDWDF technique has become of importance due to its attractive features such as massive parallelism and high accuracy inherent in nature. In this report, speed-up efficiencies in terms of achieving full parallel computing is studied for a MDWDF network of the linearized shallow water (LSWE) system important in fluid dynamics. The full parallel process in the MDWDF network is carried out based on the chained MD retiming technique, a class of software pipelining, and being implemented on a Sony PS3 exploiting the parallelism in the IBM Cell Broadband Engine (Cell/BE) to improve performance. The IBM Cell/BE containing 1 power processor element (PPE) to provide system functions and 8 synergistic processor elements (SPEs) optimized for efficient data processing perfectly fits the architecture of the retimed MDWDF model, which requires at most *1* main processor and *7* routine processors running independently. Experimental results in terms of the computational efficiency have shown that the parallelized model with 8 processors (1PPE+7SPEs) outperforms the non-parallel model only with the PPE by 4x performance advantage.

Keywords-MDWDF, physical systems, shallow water equations, full parallel computing, IBM Cell/BE.

I. INTRODUCTION

Physical system modeling is an important discipline as it enables the development of simulation engines for physically realistic processes such as fluid flow, electrical, and acoustical phenomena. The most popular kind of models for multidimensional (MD) physical systems can be represented by sets of linear and/or nonlinear partial differential equations (PDEs) with properly imposed initial and boundary conditions. Many numerical approaches existing to analyze the behavior of such physical systems include finite elements and finite differences. The finite element methods permit easy inclusion of local grid refinement and handling of complex geometries [1]. These methods, however, are computationally expensive and harder to directly and correctly set up the simulation plane than the frequently used finite difference methods. The latter approaches, on the other hand, have difficulties in handling irregular boundaries, and need local grid refinement in order to increase its measurement accuracy [2]. Furthermore, the computational load of the difference method due to local grid refinement also prevents its use for real-time hardware synthesis.

A remarkable alternative approach named wave digital filtering (WDF) network [3] built on properties of the traveling-wave formulation of lumped electrical elements to the modeling and simulation of a system of PDEs has been proposed in the past due to its excellent features that fit requirements of practical interest [4, 5]. Unlike most digital filter types, every delay element in a WDF can be interpreted physically as holding the current state of a mass or spring (or capacitor or inductors) [3]. Furthermore, because the rules for interconnecting the elementary models are based on scattering theory, all signals explicitly computed via WDF network are physically interpreted as traveling wave components of physical variables [4, 5, 14, 15].

Making use of the WDF network paradigm and analogies with electrical networks, an area rich in mathematical structure, the method, thus, draws maximum advantage of essential physical properties of such systems, in particular of causality, passivity, losslessness, stability, finite propagation velocity, etc., which can be translated to the actually considered physical problems so as to preserve important relationships between variables [4-9]. Proceeding in this way, it has the unique advantage of simultaneously offering second-order accuracy, high robustness and fault tolerance, massive parallelism, full localness (also for taking into account arbitrary boundary conditions and shapes), and explicit or at least semi-explicit computability.

Fettweis and fellow researchers [3-6] have carried out the pioneering work in the area of this subject. Our interest in the past and currently has revolved around the software toolbox development and hardware designing aspects [7-10, 16]. In particular, the full parallelism architecture of MDWDF network based on the software pipelining techniques [10] is focused on partitioned nested loop across distinct iterations in terms of spatial and temporal updating of non-parallel MDWDF network. Utilizing the full parallel architecture [10], in this study, the analysis of speed-up efficiencies in terms of achieving full parallel computing by using the chained MD retiming technique [11] is implemented on the IBM Cell Broadband Engine (Cell/BE) to improve performance for a MDWDF network of the LSWE system important in fluid dynamics. The IBM Cell/BE, which contains 1 power processor element (PPE) to provide system functions and *8* synergistic processor elements (SPEs) optimized for efficient data processing, perfectly fits the architecture of the retimed MDWDF network, which requires at most *1* main processor and *7* routine processors running independently. Experimental results have shown that the parallelized model with *8* processors (1PPE+7SPEs) outperforms the non-parallel model only with the PPE by 4x performance advantage.

II. SUMMARY OF SWE SYSTEM AND MODELING TECHNIQUES

A. Shallow Water System and Its Corresponding Multidimensional Wave Digital Filtering Network

Let us consider a linearized shallow water (LSWE) system characterized by a set of PDE for the surface displacement, η [1, 7, 12]:

$$\begin{cases} \dfrac{\partial v_1}{\partial t} - fv_2 + g\dfrac{\partial \eta}{\partial x} = 0 \\ \dfrac{\partial v_2}{\partial t} + fv_1 + g\dfrac{\partial \eta}{\partial y} = 0 \\ H\dfrac{\partial v_1}{\partial x} + H\dfrac{\partial v_2}{\partial y} + \dfrac{\partial \eta}{\partial t} = 0 \end{cases} \tag{2.1}$$

Here the horizontal velocities v_1 and v_2 are directed along the spatial domain of *x* and *y* direction, respectively. Furthermore, *h* is the total water depth defined as the sum of the undisturbed water depth (a constant mean depth) *H* and the free surface elevation η measured upward from the undisturbed surface, i.e. $h=H+\eta$. The gravity acceleration *g* and the Coriolis parameter *f* are constants. The hydrostatic formulation is the combined physical processes of radiation, refraction, diffraction and reflection when the system is confined in a bounded domain with non-empty piecewise boundary. In an unbounded domain, the LSWE system, however, generates divergent and non-reflective waves based on some artificial open boundaries [2]. As a result, this modeling is quite successful in predicting the dynamics of the surface gravity waves, and is usually applied to the global modeling on large scale oceanographic or atmospheric quantities like transports and surface elevation, Tsunami modeling and simulation [2, 14].

Applying the standard procedure known from MD wave digital filtering [4, 5] for transforming the PDE to its equivalent discrete passive model, a MDWDF network depicted in Fig. 1(a) is obtained for the numerical integration of the LSWE system. Since the resulting network behaves in the same way as the continuous one, it also preserves passivity of the discrete dynamical system, thus ensuring full robustness and stability of the algorithm [4]. Details of converting the given physical system to form the MDWDF network via a lumped MD-passive Kirchhoff circuit can be found in [7].

B. Chained Multidimensional Retiming

The objective of the chained MD retiming technique is to legally change the delay of edges in an MD data-flow graph (MDFG) such that nonzero delays on all edges can be obtained in order to achieve the full parallelism. We note that the form of MDFG is described as a cyclic data flow graph with the tuple *(V, E, D, t)* to represent a node-weighted and edge-weighted graph where *V* is the set of computation nodes in the loop body, *E* is the set of directed edges representing the dependence between two nodes, *d* is a function representing the MD-delay between two nodes, and *t* is the time required for computing a certain node [11, 12].

Define a scheduling subspace of a realizable MDFG by $S = \{s: d(e) \cdot s \geq 0, \forall e \in E\}$. The technique of the chained MD retiming [11, 12] for scheduling MD problems starts with finding a schedule vector $s \in S^{+}$ where S^{+} is a strictly positive scheduling subspace defined by

$$S^{+} = \{s \in S : d(e) \cdot s > 0, \forall d(e) \neq (0, \cdots, 0), e \in E\}.$$

Given any scheduling element *s*, a legal retiming function *r* of a realizable MDFG for each node of a loop body can be obtained when it is orthogonal to *s* according to [7, 11]. As a consequence, the chained MD scheduling generates a retiming vector *r(u)*for each computing node u in the MDFG such that the MD-delay is pushed from incoming edges of *u* to its outgoing edges and new delay of each edge is given by

$$d_r(e_j) = \begin{cases} d(e_j) + r(u) & \text{all outgoing edges } e_i \text{ from } u \\ d(e_j) - r(u) & \text{all incoming edges } e_j \text{ of } u \end{cases} \tag{2.2}$$

It is important to note that following the principle of MD retiming [7, 11], *r*(*u*) can be obtained without difficulty by the following formula

$$r(u) = (K_n - i) \cdot r \tag{2.3}$$

where K_n is the maximum length of the existing chains with the highest level number *n* in the construction process of a MDFG, while *i* is the level number of the node *u*. If an incoming edge has zero delay, it is necessary to apply the same retiming function to that incoming node and to leave the sum of delays of the loop body unchanged.

Provided by a MDFG of the MDWDF network in Fig. 1(b) where each operation of the network loop body as detailed in Fig. 1(c) is executed in one clock cycle by one time unit, a schedule table in Tab. 1(a) can be obtained, which clearly shows that the traverse of an iteration of the loop requires a minimum of *7* clock cycles. Having the choice of the maximum length of the existing chain K_n*=5* (K_n*=7*, respectively) with the highest level *n=4* (*n=6*, respectively) for the left loop body (the right loop body, respectively) of the MDFG, Eq. (2.3) yields the retiming vectors listed in Tab. 2 for each node in the MDFG. Substituting the retiming vector into Eq. (2.2), a full retimed MDFG of Fig. 2(a) is built and its corresponding MDFG loop body in Fig. 2(b) and schedule table as listed in Tab. 1(b) for each iteration is analyzed. As compared to the level of clock cycle in Tab. 1(a), analytical results listed in Tab. 1(b) have simply demonstrated the achievement of full parallel computation in the retimed MDFG of Fig. 2(a). In view of the retimed MDFG, all edges contain non-zero delay, which eliminates the intra-iteration dependencies. Furthermore, the total delays of each loop remain unchanged.

To achieve the full parallelism, clearly the left loop body must requires at most *5* parallel processors synchronously executing all operations, while at most *7* parallel processors are required for the right loop body. The loop body of the retimed MDFG illustrated in Fig. 2(b) containing prologue and epilogue processes is executed to provide the necessary data for the parallel loops, which complementarily completes the process. As a result, provided by at most *7* parallel processors named $P_1,\cdots, P_7$, the retimed MDFG has achieved significantly reducing the number of necessary clock cycles into one pass.

III. Hardware Experiment

In this section, we present simulation results of the LSWE system implemented on the IBM Cell/BE PS3 with architecture depicted in Fig. 3 for the computational efficiency of the fully parallelized MDWDF network against the non-parallel one. The parallelized model based on the retimed MDFG of Fig. 2(b) is carried out by 8 processors (1PPE+7SPEs), while the non-parallel model based on Fig. 2(b) is implemented only by the PPE itself. In particular of interest, with the help of memory flow controller-direct memory access (MFC-DMA) queue the 8 processors, $P_0,\ldots P_7$, in the loop body of the retimed MDFG in Fig. 3 are assigned respectively to the IBM cell /BE PPE,SPE1,…,SPE7 on PS3. As a result, the parallelism in the left body loop renders 5 SPEs, i.e. SPE1,…,SPE5, while the parallelism in the right body loop is tackled by 7 SPEs, i.e. SPE1,…,SPE7. Apart from the parallel process implemented on the SPUs, there are processes of prologue, epilogue, and program initiation, which are controlled fully by the PPU. Details of the cells arrangement for the process of the retimed loop operations can be seen in Fig. 5, which depicts the 64-bit power architecture for the parallel MDWDF model.

Having comparison of performance based on 4 groups of iteration temporal points obtained for both models, i.e. 25, 50, 100, and 200 as listed in Tab. 3, it is expected that the maximum computation time will be doubled when the temporal points are increased by twice. On one hand, Fig. 4(a) shows results of the non-parallel model, which has clearly illustrated that the computational time is nearly doubled as also indicated in Tab. 3. On the other hand, Fig. 4(b), depicts excellent results of the parallelized model, which although shows the similar property as Fig. 4(a) does, but in a more efficiency matter. More precisely, Tab. 3 illustrates that the parallel model running independently on 8 processors is 4x more efficient (runtime speedup) than the non-parallel model running only on a single processor. The key performance advantage comes from utilizing the 8 de-coupled SPE SIMD engines with dedicated resources including large register files and DMA channels.

IV. Conclusion

In this study, we have reported the excellent computational performance of the parallel MDWDF network model running on the IBM cell/BE. The full parallelism of the MDWDF network was carried out by working together 8 cells of the cell broadband engine. In particular of interest, the PPE controls the prologue, epilogue and program initiation, while the parallel process of the loop body is rendered at most by 7 PPEs. Simulation results showed that the runtime speed of the parallel model outperforms that of the non-parallel model by 4x of computational time.

REFERENCES

[1] Kundu , P. K. (1990), Fluid Mechanics, Academic Press Limited, London.

[2] Kowalik, Z. and Murty, T. S. (1993), Numerical Modelling of Ocean Dynamics, World Scientific Publishing.

[3] Fettweis, A. (1986), “Wave-digital filters: Theory and practice,” Proceedings of IEEE, 74:270-327.

[4] Fettweis, A. and Nitsche, G. (1991), “Transformation approach to numerical integrating PDEs by means of WDF principles,” Multidimensional Systems Sig. Proc. 2:127-159.

[5] Fettweis, A. and Nitsche G. (1991), “Numerical integration of partial differential equations using principles of multidimensional wave-digital filters,” Journal of VLSI Signal Processing, 3:7-24.

[6] Krauss, H., Rabenstein. R., and Gerken. M. (1996), “Simulation of wave propagation by multidimensional digital filters,” Journal of Simulation Practice and Theory, 4:361-382.

[7] Tseng, C. H. and Lawson, S. S. (2003), “Discrete modelling of shallow water equations using the multidimensional wave digital filters,” European Conference in Circuit Theory and Design (ECCTD), Poland.

[8] Tseng, C. H. and Lawson, S. S. (2004), “Initial and boundary conditions in multidimensional wave digital filter algorithms for plate vibration,” IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications, 51(8):1648-1663.

[9] Tseng, C. H. (2012), “Modelling and visualization of a time-dependent shallow water system using nonlinear Kirchhoff circuit,” IEEE Transactions on Circuits and Systems-I: Regular papers, 59(6):1265-1277.

[10] Tseng, C. H. and Lawson, S. S (2004), “Full parallel process for multidimensional wave digital filtering via multidimensional retiming technique,” ISCAS, III:209-212, Vancouver, Canada.

[11] Passos, N. L. and Sha, E. H-M (1996), “Achieving full parallelism using multi-dimensional retiming,” IEEE Trans. Parallel and Distributed Systems, 7(11):1150-1163.

[12] Passos, N. L. and Sha, E. H-M (1994), “Full parallelism in uniform nested loops using multidimensional retiming,” Proceedings of the International Conference on Parallel Processing, Saint Charles, IL, II:130-133.

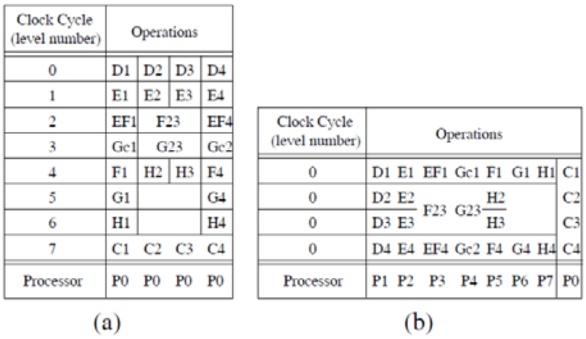
[13] Hanert, E., Roux, D. Le, and Legat, V. (2005), “An efficient eulerian finite element method for the shallow water equations,” Ocean Modeling, 10: 115-136.

[14] Tseng, C. H., Dao, N-D-T, and Chang, C. C. (2010), “Modeling and visualizing seismic wave propagation in elastic medium using multi-dimension wave digital filtering approach,” Journal of World Academy of Science, Engineering and Technology, 69:424-430.

[15] Tseng, C. H. (2012), “Numerical stability verification of a two-dimensional time-dependent nonlinear shallow water system using multidimensional wave digital filtering network,” Springer-Circuits, Systems and Signal Processing , DOI:10.1007/s00034-012-9461-7.

[16] Tseng, C. H. and Jeang, Y. L. (2012), FPGA Digital IC Design and Practice: use Verilog HDL and Xilinx ISE, 1st edition, Tsang Hai Book Publishing Co., ISBN 978-986-6184-85-7. In Chinese

Tab 1. Schedule tables: (a) MDFG. (b) a retimed MDFG.



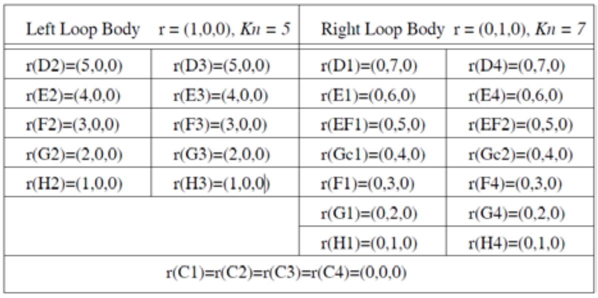
Clock Cycle (level number)	Operations			
0	D1	D2	D3	D4
1	E1	E2	E3	E4
2	EF1	F23		EF4
3	Gc1	G23		Gc2
4	F1	H2	H3	F4
5	G1			G4
6	H1			H4
7	C1	C2	C3	C4
Processor	P0	P0	P0	P0

(a)

Clock Cycle (level number)	Operations							
0	D1	E1	EF1	Gc1	F1	G1	H1	C1
0	D2	E2	F23	G23		H2		C2
0	D3	E3				H3		C3
0	D4	E4	EF4	Gc2	F4	G4	H4	C4
Processor	P1	P2	P3	P4	P5	P6	P7	P0

(b)

Tab 2. Retiming vector for each node in the MDFG



Left Loop Body r = (1,0,0), *Kn* = 5		Right Loop Body r = (0,1,0), *Kn* = 7	
r(D2)=(5,0,0)	r(D3)=(5,0,0)	r(D1)=(0,7,0)	r(D4)=(0,7,0)
r(E2)=(4,0,0)	r(E3)=(4,0,0)	r(E1)=(0,6,0)	r(E4)=(0,6,0)
r(F2)=(3,0,0)	r(F3)=(3,0,0)	r(EF1)=(0,5,0)	r(EF2)=(0,5,0)
r(G2)=(2,0,0)	r(G3)=(2,0,0)	r(Gc1)=(0,4,0)	r(Gc2)=(0,4,0)
r(H2)=(1,0,0)	r(H3)=(1,0,0)	r(F1)=(0,3,0)	r(F4)=(0,3,0)
		r(G1)=(0,2,0)	r(G4)=(0,2,0)
		r(H1)=(0,1,0)	r(H4)=(0,1,0)
r(C1)=r(C2)=r(C3)=r(C4)=(0,0,0)			

Tab 3. Maximum computation time for 4 temporal groups.

Maxi. Computation Time (sec)		Simulated Time Step			
		25	50	100	200
Model Type	Non-parallel	6.58	15.03	28.72	54.74
	Parallel	1.61	3.26	6.60	13.28
performance advantage		4.09	4.61	4.35	4.12

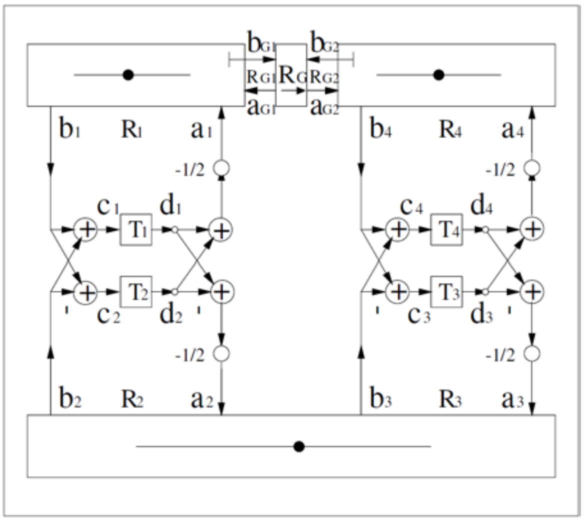


Fig. 1(a) The MDWDF network of the SWE system.

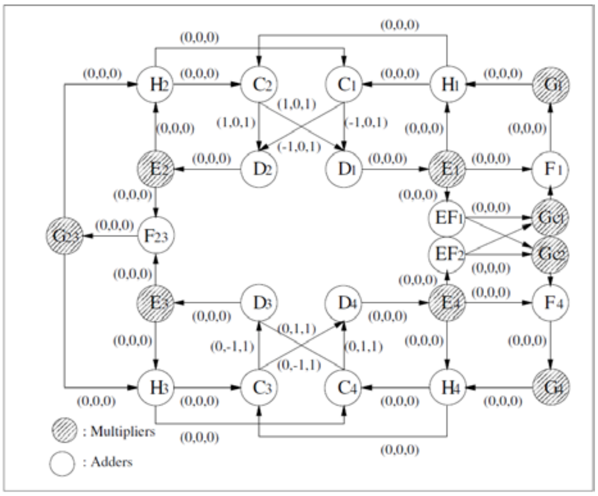
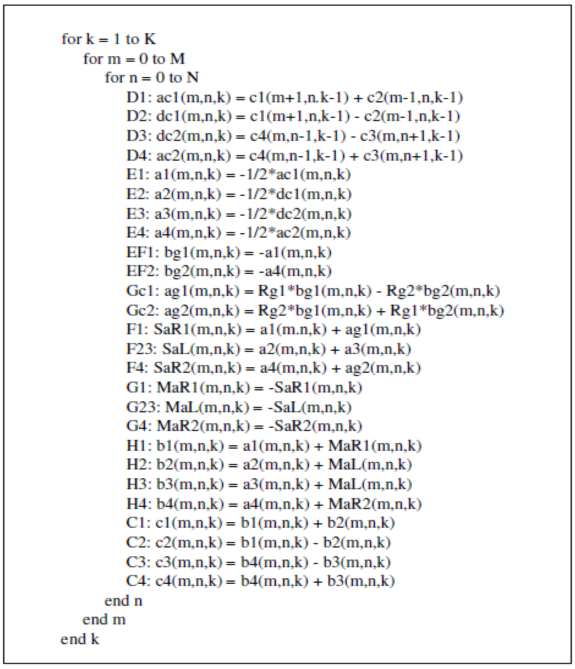


Fig. 1(b) A MDFG representing the MDWDF network.



```
for k = 1 to K
    for m = 0 to M
        for n = 0 to N
            D1: ac1(m,n,k) = c1(m+1,n.k-1) + c2(m-1,n,k-1)
            D2: dc1(m,n,k) = c1(m+1,n,k-1) - c2(m-1,n,k-1)
            D3: dc2(m,n,k) = c4(m,n-1,k-1) - c3(m,n+1,k-1)
            D4: ac2(m,n,k) = c4(m,n-1,k-1) + c3(m,n+1,k-1)
            E1: a1(m,n,k) = -1/2*ac1(m,n,k)
            E2: a2(m,n,k) = -1/2*dc1(m,n,k)
            E3: a3(m,n,k) = -1/2*dc2(m,n,k)
            E4: a4(m,n,k) = -1/2*ac2(m,n,k)
            EF1: bg1(m,n,k) = -a1(m,n,k)
            EF2: bg2(m,n,k) = -a4(m,n,k)
            Gc1: ag1(m,n,k) = Rg1*bg1(m,n,k) - Rg2*bg2(m,n,k)
            Gc2: ag2(m,n,k) = Rg2*bg1(m,n,k) + Rg1*bg2(m,n,k)
            F1: SaR1(m,n,k) = a1(m.n,k) + ag1(m,n,k)
            F23: SaL(m,n,k) = a2(m,n,k) + a3(m,n,k)
            F4: SaR2(m,n,k) = a4(m,n,k) + ag2(m,n,k)
            G1: MaR1(m,n,k) = -SaR1(m,n,k)
            G23: MaL(m,n,k) = -SaL(m,n,k)
            G4: MaR2(m,n,k) = -SaR2(m,n,k)
            H1: b1(m,n,k) = a1(m,n,k) + MaR1(m,n,k)
            H2: b2(m,n,k) = a2(m,n,k) + MaL(m,n,k)
            H3: b3(m,n,k) = a3(m,n,k) + MaL(m,n,k)
            H4: b4(m,n,k) = a4(m,n,k) + MaR2(m,n,k)
            C1: c1(m,n,k) = b1(m,n,k) + b2(m,n,k)
            C2: c2(m,n,k) = b1(m,n,k) - b2(m,n,k)
            C3: c3(m,n,k) = b4(m,n,k) - b3(m,n,k)
            C4: c4(m,n,k) = b4(m,n,k) + b3(m,n,k)
        end n
    end m
end k
```

Fig. 1(c) The corresponding loop body of the MDFG.

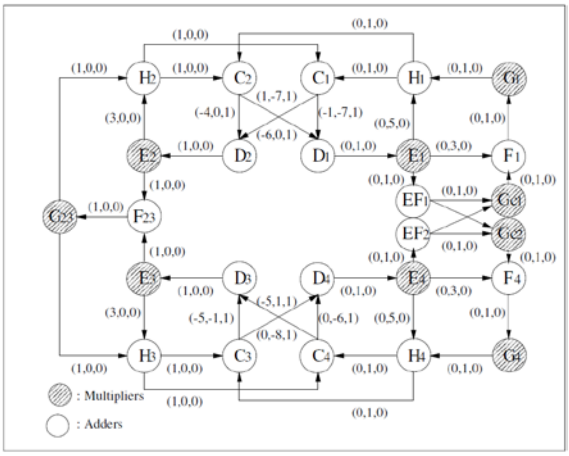


Fig. 2(a) A retimed MDFG of the MDWDF network.

```
for k = 1 to K
   for n = 0 to N      % left loop body
      --- prologue ---
      for m = 0 to M-5
P1 ↗ D2: dc1(m+5,n,k) = c1(m+6,n,k-1) - c2(m+4,n,k-1)
   ↘ D3: dc2(m+5,n,k) = c4(m+5,n-1,k-1) - c3(m+5,n+1,k-1)
P2 ↗ E2: a2(m+4,n,k) = -1/2*dc1(m+4,n,k)
   ↘ E3: a3(m+4,n,k) = -1/2*dc2(m+4,n,k)
P3 → F23: SaL(m+3,n,k) = a2(m+3,n,k) + a3(m+3,n,k)
P4 → G23: MaL(m+2,n,k) = -SaL(m+2,n,k)
P5 ↗ H2: b2(m+1,n,k) = a2(m+1,n,k) + MaL(m+1,n,k)
   ↘ H3: b3(m+1,n,k) = a3(m+1,m,k) + MaL(m+1,n,k)
      end m
      --- epilogue ---
   end n
   for m = 0 to M    % right loop body
      --- prologue ---
      for n = 0 to N-7
P1 ↗ D1: ac1(m,n+7,k) = c1(m+1,n+7.k-1) + c2(m-1,n+7,k-1)
   ↘ D4: ac2(m,n+7,k) = c4(m,n+6,k-1) + c3(m,n+8,k-1)
P2 ↗ E1: a1(m,n+6,k) = -1/2*ac1(m,n+6,k)
   ↘ E4: a4(m,n+6,k) = -1/2*ac2(m,n+6,k)
P3 ↗ EF1: bg1(m,n+5,k) = -a1(m,n+5,k)
   ↘ EF2: bg2(m,n+5,k) = -a4(m,n+5,k)
P4 ↗ Gc1: ag1(m,n+4,k) = Rg1*bg1(m,n+4,k) - Rg2*bg2(m,n+4,k)
   ↘ Gc2: ag2(m,n+4,k) = Rg2*bg1(m,n+4,k) + Rg1*bg2(m,n+4,k)
P5 ↗ F1: SaR1(m,n+3,k) = a1(m.n+3,k) + ag1(m,n+3,k)
   ↘ F4: SaR2(m,n+3,k) = a4(m,n+3,k) + ag2(m,n+3,k)
P6 ↗ G1: MaR1(m,n+2,k) = -SaR1(m,n+2,k)
   ↘ G4: MaR2(m,n+2,k) = -SaR2(m,n+2,k)
P7 ↗ H1: b1(m,n+1,k) = a1(m,n+1,k) + MaR1(m,n+1,k)
   ↘ H4: b4(m,n+1,k) = a4(m,n+1,k) + MaR2(m,n+1,k)
      end n
      --- epilogue ---
   end m
   for m = 0 to M
      for n = 0 to N
         C1: c1(m,n,k) = b1(m,n,k) + b2(m,n,k)
         C2: c2(m,n,k) = b1(m,n,k) - b2(m,n,k)
         C3: c3(m,n,k) = b4(m,n,k) - b3(m,n,k)
         C4: c4(m,n,k) = b4(m,n,k) + b3(m,n,k)
      end n
   end m
end k
```

Fig. 2(b) The corresponding loop body of the retimed MDFG.

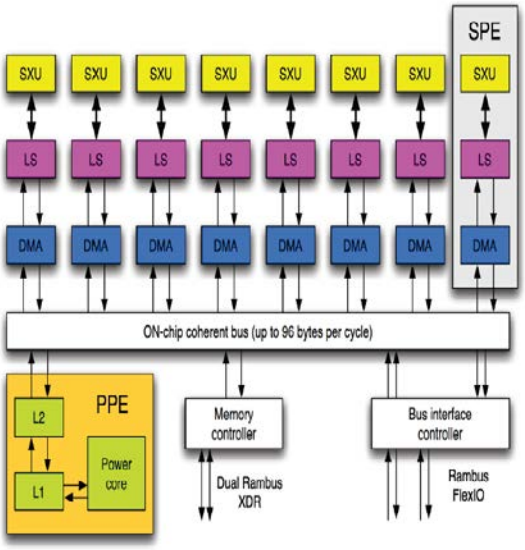


Fig. 3 IBM cell broadband engine schema with one PPE and eight SPEs.

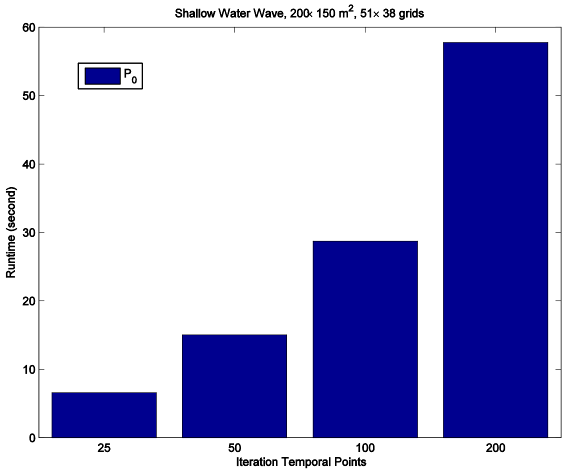


Fig. 4(a) Computational time of the non-parallel MDWDF network model.

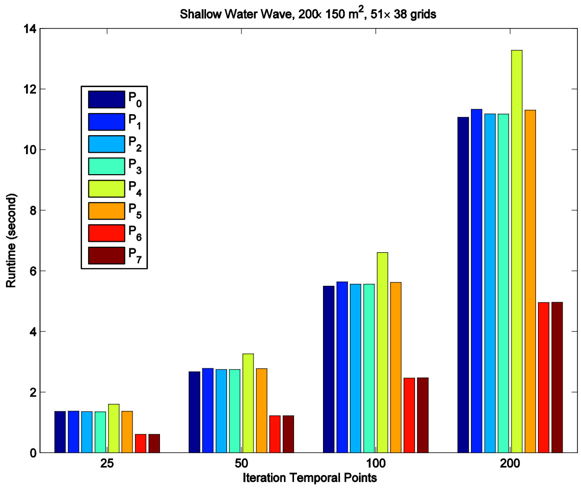


Fig. 4(b) Computational time of the parallelized MDWDF network model.

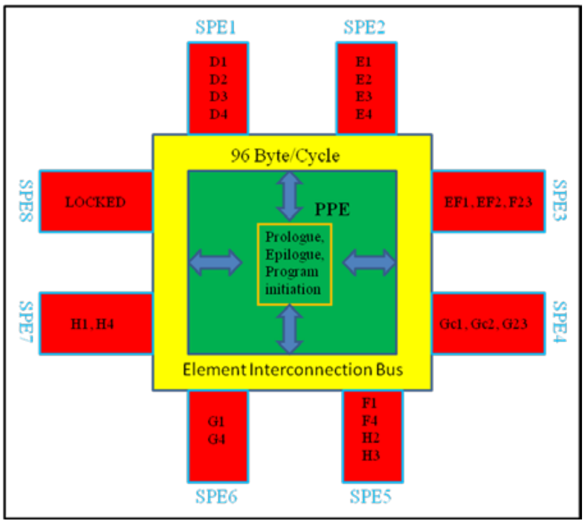


Fig. 5 A custom designed 64-bit power architecture for the parallel MDWDF network model.