

실시간 ALU 에러 검출의 에러 전파 문제 해결을 위한 효과적인 Berger Code Prediction 하드웨어 구조

The ALU Berger Code Prediction Hardware for Error Propagation Problem in Concurrent ALU Error Detection

김근배, 김일웅, 강일권, 박기현, 양동훈, 강성호

연세대학교 전기전자공학과

{kbg9572, woong, neokings, cimabear, ysydh}@soc.yonsei.ac.kr, shkang@yonsei.ac.kr

Abstract

In this paper, an enhanced Berger Code Prediction (BCP) hardware architecture is proposed. To eliminate the error propagation of the previous BCP checker and minimize additional overhead, a dedicated internal carry vector generator and zero counting logic are newly designed. The hardware implementation results show the proposed architecture solves error propagation problem with minimum additional overhead.

I. 서론

Data-path내에 존재하는 ALU는 processor의 성능을 결정하는 여러 요소 중 가장 중요한 부분이며 이러한 이유로 동작 중 발생 가능한 ALU의 error를 검출하기 위한 다양한 방법들이 개발되어 왔다. 실시간 error 검출 방법은 결국 추가의 redundancy를 이용한 방법을 통하여 구현되는데 하드웨어적으로 동일 ALU를 복수 개 사용하고 이들의 결과를 분석하는 방법이 있으나 이는 중복되는 ALU의 개수만큼의 하드웨어 증가를 유발한다, 따라서 추가적인 check bit과 이를 검사하는 추가 하드웨어를 내장하는 방법이 주로 개발되어 왔고 그 중 한 예가 Berger Code를 이용한 방법이다 [1-4]. Berger Code Prediction 방법은 ALU의 logic, arithmetic 연산에 사용되는 operand의 1 또는 0의 개수를 이용, 대상 ALU의 연산과 동일한 연산을 수행하고 이를 최종 결과값의 1 또는 0의 개수와 비교하여 error의 유, 무를 검사하는 방식이다. 이 방법은 간단한 BCP symbol을 통하여 모든 unidirectional error를 검출할 수 있다는 점에서 효과적이지만 하드웨어 구현과정에서 기존 구조들이 채택한 방법은 arithmetic operation에서 사용되는 internal carry vector를 검사 대상 ALU가 제공할 수 있어야 하며 더

불어 대상 ALU에서 error가 발생하였을 경우 해당 error가 internal carry vector를 통하여 checker에 전달, 예측 연산을 수행하므로 잘못된 검사를 수행하게 되는 구조적인 문제를 가지게 된다.

II. 본론

본 논문은 기존 BCP checker의 error propagation 문제를 해결하고 추가 오버헤드를 최소화하기 위하여 새로운 internal carry generator를 개발하였다. 제안하는 BCP checker의 전체 구조는 그림 1과 같다.

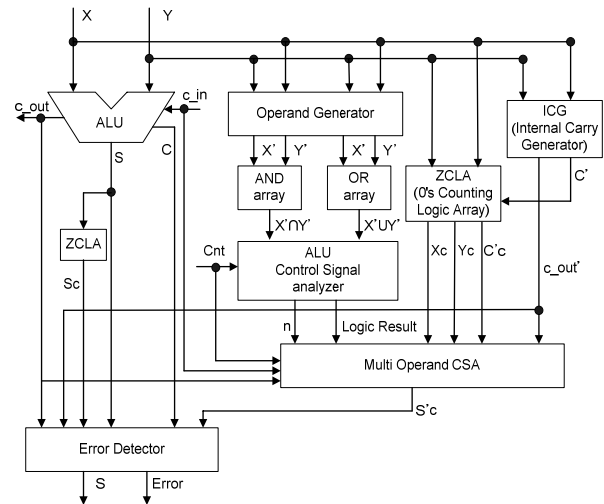


그림 1. 제안하는 BCP checker 구조

제안하는 BCP checker 구조는 ALU 연산에 사용되는 operand와 control signal을 동시에 입력으로 받으며 control signal의 값에 따라서 ALU control signal analyzer에서 해당 logic result를 생성하고 arithmetic 연산일 경우는 내부 internal carry generator의 값을 함께 이용하여 BCP symbol을 생성한다. 제안하는 구조를 위하여 개발된 internal carry generator의 구조는 그림 2와 같다.

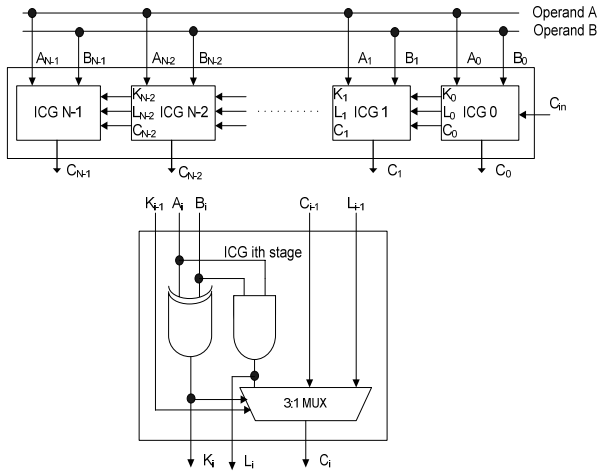


그림 2. 제안하는 internal carry generator 구조

제안된 internal carry generator의 동작은 다음과 같다. $K_i = A_i \oplus B_i$ (A exclusive OR B)는 두 operand의 bit의 값의 동일 여부를 검사하며 $K_i = 0$ 일 때 $A_i = B_i = 1$ 또는 $A_i = B_i = 0$ 을 가지는 데 이 경우 입력으로 들어오는 C_{in} 의 값에 영향을 받지 않고 carry를 생성하게 되며 $C_i = L_i = A_i \wedge B_i$ 가 되어 곧바로 carry를 생성한다. $K_i = 1$ 인 경우는 이전 $i-1$ 번째의 carry값에 영향을 받으므로 C_i 의 값을 알아야 하지만 K_{i-1} 의 값이 0이면 $C_i = L_{i-1}$ 이 되므로 carry전파의 영향을 줄일 수 있다. 만일 $K_i = K_{i-1} = 1$ 이면 C_{i-1} 의 연산을 통한 결과값을 이용하게 된다. 하지만 K_i 값이 0이 되는 곳이 발생할 경우 carry는 바로 생성되기 때문에 최악의 경우 모든 조합이 $K_i = 1$ 인 경우를 제외하면 마지막 bit조합의 carry 생성까지 영향이 전파되지 않게 된다.

III. Hardware 구현 결과

제안된 하드웨어는 TSMC 0.25 μ m 공정 파일과 SYNOPSIS design compiler를 이용하여 구현하였다. 표 1은 제안된 하드웨어의 주요 모듈의 하드웨어 overhead (H, gate count)와 timing overhead (T, nano second)의 값을 32 bit operand에 대해서 구현된 결과를 토대로 정리한 것이다.

표 1. 제안하는 BCP checker의 구현 결과

Bit Count	ALUCA		ZCLA		ICG		MCSA		Error Detector	
	H	T	H	T	H	T	H	T	H	T
32	159	3.7	179	6.2	305	3.4	584	4.2	50	1.4

제안하는 BCP checker 구조는 필연적으로 internal carry generator 모듈 추가로 인하여 하드웨어

overhead의 증가가 발생되지만 기존 구조가 ALU 연산이 끝난 이후 internal carry vector를 받기 때문에 timing overhead는 오히려 기존 구조에 비해 개선된다. 또한 ICG의 추가로 인하여 ALU로부터의 error propagation 문제가 해결될 수 있게 된다.

IV. 결론

제안하는 BCP checker 구조는 실시간 ALU 검사를 위하여 사용되는 기존의 BCP 하드웨어 구조가 ALU 내부의 internal carry vector를 재사용함으로써 발생하는 error propagation 문제를 해결하기 위하여 BCP checker 내부에 별도의 internal carry generator를 내장하였으며 이를 통한 하드웨어 및 timing overhead가 최소화 되기 위한 구조를 개발하였다. 제안된 구조는 필연적으로 기존 BCP checker와 비교하여 하드웨어 overhead의 증가를 가져오지만 추가적인 timing overhead 없이 error propagation 문제를 해결한 점에서 실시간 ALU 검출 방식의 성능 향상을 달성할 수 있었다.

Acknowledgement 본 논문은 IDEEC (IC Design Education Center)의 CAD tool 지원을 받은 것이다.

참고문헌

- [1] Lo, J.-C., Thanawastien, S., Rao, T.R.N., Nicolaidis, M., "An SFS Berger check prediction ALU and its application to self-checking processor designs," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume 11, Issue 4, pp. 525 – 540, April 1992.
- [2] Morozov, A., Saposhnikov, V.V., Saposhnikov, V.I., Gossel, M., "New self-checking circuits by use of Berger-codes," Proceedings of 6th IEEE International On-Line Testing Workshop, pp. 141 – 146, July 2000.
- [3] Russell, G., Maamar, A.H., "Check bit prediction scheme using Dong's code for concurrent error detection in VLSI processors," IEE Proceedings Computers and Digital Techniques, Volume 147, Issue 6, pp. 467 – 471, Nov. 2000.
- [4] Stankovic, T.R., Stojcev, M.K., Djordjevic, G.L., "Design of self-checking combinational circuits," Proceedings of 6th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Service, Volume 2, pp. 763 – 768, 2003.