

論文2001-38SD-1-7

P/G블록을 가진 ALU에서 글리치 전파제거에 의한 저전력 실현

(A Low Power Realization by Eliminating Glitch-Propagation in an ALU with P/G blocks)

柳 凡 善 *, 李 成 鉉 *, 李 奇 榮 *, 趙 泰 元 *

(Beom Seon Ryu, Sung Hyun Lee, Kie Young Lee, and Tae Won Cho)

요 약

본 논문에서는 기존의 P(캐리전파)/G(캐리발생) 블록을 가진 ALU구조에서 발생하는 글리칭 전력소모를 최소화시킨 새로운 구조에 대해서 기술한다. 일반적으로 회로에서 발생하는 많은 글리치가 다음 단 회로로 전파될 때, 필요 없이 많은 전력소모가 발생된다. 따라서 본 논문에서는 ALU의 P/G 블록에서 발생하는 글리치를 제거하는 구조를 제안하였다. P/G블록에서 글리치가 제거되면 다음 단인 Sum 발생 블록에서 글리치에 의한 신호전이가 줄어들고, 이에 따라 전력소모가 줄어든다. P/G 블록의 출력 단에 발생하는 글리치 제거를 위해, 기존의 P/G블록내에 래치를 삽입하였다. 래치의 인에이블 신호는 P/G블록의 출력 인에이블 시간을 제어함으로써, P/G블록의 출력 단의 글리치를 제거시키는 역할을 한다. 16비트 ALU를 구현하여 HSPICE로 모의 실험한 결과, 제안한 구조는 지연시간의 증가가 거의 없으면서 약 28%의 글리칭 전력소모가 감소되었다.

Abstract

This paper presents a new ALU architecture to minimize glitching power consumption which is appeared in the conventional one with P(carry propagation)/G(carry generation) blocks. In general, A lot of glitches generated once are propagating into the next stage of circuits to make unnecessary power dissipation. Therefore, a new ALU architecture which removes the glitches at the output of P/G blocks is presented in this paper. If a lot of glitches at the output of P/G blocks are removed, then the signal transitions caused by glitches are reduced in the sum generation block and hence power consumption is also reduced. A latch is inserted into the conventional P/G blocks to remove the glitches at the output of P/G blocks. Latch enable signal can make a role in eliminating a lot of glitches at the P/G's outputs by controlling output enable time. Experimental results from HSPICE simulations with implementing 16-b ALU show 28% reduction in glitching power consumption with negligible delay penalty.

I. 서 론

이동 통신의 발달 및 집적회로의 집적도가 높아짐에

따라서 저전력 소모는 회로설계자들에게 가장 큰 설계 목표로 대두되었다^{[1]-[3]}. 구체적인 저전력 회로설계의 필요성으로는 회로의 신뢰성 향상, 고성능 시스템의 패키징과 냉각장치로 인한 비용 절감 및 휴대용 전자기에 사용되는 배터리의 수명연장 등이다^{[1]-[3]}. ALU (Arithmetic Logic Unit)는 마이크로 프로세서나 DSP(Digital Signal Processor) 시스템에서 실행유닛으로 사용되는 중요한 데이터 패스 블록중의 하나이다

* 正會員, 忠北大學校 工科大学 電氣電子工學部
(School of Electronic and Electrical Engineering,
Chungbuk National University)

接受日字:2000年7月19日, 수정완료일:2000年12月13日

[4-5]. ALU는 덧셈기와 비교하여 설명할 때, 일종의 수정된 덧셈기 구조라고 한다^[6]. 그 이유는 덧셈기는 캐리전파신호(carry propagation signal)와 캐리발생신호(carry generation signal)를 계산하는 하드웨어가 XOR 및 AND게이트로 고정되어 있는 반면에, ALU는 연산 코드(op-code)에 따라서 캐리전파신호 및 캐리발생신호 값이 달라지기 때문에, P/G블록의 출력 값이 연산 코드에 따라서 프로그램 될 수 있는 NMOS 패스 트랜지스터 회로망으로 구성된다^[5-8,11,12].

지금까지 발표된 ALU구조는 저전력 관점보다는 하드웨어의 최소화 및 고속연산을 위한 구조 및 회로가 대부분이었으며^[9-11], 보고된 논문중에서 [12]가 저전력에 관한 연구로써 논리연산시에 덧셈기 블록을 디스에이블시켜 전력소모를 줄인 구조이다. 본 논문에서는 기존의 ALU구조에서 P/G블록에서 발생하는 글리치 및 이렇게 발생된 글리치들이 다음 단으로의 전파제거를 통하여, 글리칭 전력소모를 최소화시킨 ALU의 구조 및 회로구현에 대해서 기술한다. 기존의 ALU구조에서 나타나는 글리치 발생의 원인은, P/G블록에 인가되는 두 종류의 입력신호들의 도착시간의 차이 때문이다^[5-8]. P/G블록의 입력으로는 4비트의 데이터 신호와, 4비트의 제어신호가 있다. 데이터 신호는 입력 레지스터로부터 직접 P/G블록으로 도착하는데 비하여, 제어신호는 연산코드가 ALU 제어기(ALU controller)를 통해서 해독된 신호로 P/G블록에 입력된다^[4]. 따라서 P/G블록에 인가되는 데이터 신호와 제어신호 사이에는 ALU 제어기만큼의 지연시간 차이가 발생되며, 이것이 P/G블록의 출력 단에 많은 글리치를 발생시킨다. 그러나 문제는 이렇게 발생된 글리치들이 다음 단의 데이터 패스 블록으로 전파됨으로써, 더 큰 전력소모가 발생하는데 있다. 일반적으로 글리치가 전력소모에 미치는 영향을 분석하면, 글리치 발생 자체보다는 발생된 글리치들이 다음 단 회로로 전파되는 경우가 더 큰 영향을 미친다^[13-15].

본 논문에서는 기존의 글리치 발생 및 전파문제를 해결하기 위하여, 기존의 P/G블록안에 인에이블 신호를 갖는 래치를 삽입하였다. 글리치 제거를 위한 일반적인 방법중의 하나로써, 글리치 신호가 발생되는 노드에 래치를 삽입하는 방법이 있는데^[15] 이것은 다음과 같은 두 가지 단점을 가진다. 첫째로는 래치의 삽입으로 인한 과도한 지연시간의 증가이고^[16], 둘째로는 요구되는 래치의 인에이블 신호의 타이밍이 오직 한 시점

에서 결정되는 점이다. 그런데 기존의 P/G블록의 출력 단 회로구조가 이미 래치와 유사한 형태로 되어 있어서, 일반적인 래치삽입에 따른 문제점들이 완화되거나 또는 해결된다. 구체적으로 래치삽입에 의한 발생하는 과도한 지연시간 증가는, 기존의 P/G블록^[5-8]의 출력형태가 래치와 유사한 형태로 되어있기 때문에, 제안한 P/G블록에 래치삽입에 의한 하드웨어 오버헤드는 NMOS 패스 트랜지스터뿐이다. 이로 인하여 기존의 P/G블록에 비교하여 제안한 P/G블록의 지연시간 증가가 최소화된다. 두 번째로 P/G블록내의 래치의 출력이나, 직접 Sum블록에 연결되지 않고 구동회로를 거쳐서 연결되기 때문에, 요구되는 래치의 인에이블 신호의 타이밍이 오직 한 시점이 아니라 유한한 시간구간이 존재한다. 이것은 글리치 필터링(glitch filtering)에 의한 현상으로^[2,17,18], 래치의 출력에 작은 폭의 글리치가 존재하더라도 다음 단의 구동회로를 통과하면서 글리치의 폭이 감소되고, 최종적으로는 글리치의 전압레벨(글리치의 진폭)이 $1/2V_{dd}$ 보다 작아지면서 최종출력에서 글리치가 제거된다. 이러한 글리치 필터링 현상 때문에 제안한 P/G블록의 출력 단에 글리치가 제거되는 상황에서, 요구되는 인에이블 신호의 타이밍은 최소값 및 최대값을 갖는다. 이것은 인에이블 신호가 유한한 시간구간을 갖음으로써, 공정변화에 대하여 안정된 회로성을 나타낸다.

일반적으로 글리치는 조합논리회로에서 두 개 이상의 입력신호의 도착시간 차이가 발생함으로써, 출력 단에 과도상태에서 나타나는 원하지 않는 파형이다^[13-19]. 글리치 발생은 회로구조 및 입력신호의 통계적 특성과 밀접한 관계가 있으며, 일반적으로 보고되기로는 글리칭 전력소모가 전체 전력소모의 10~60%까지 보고되는 바^[19], 글리칭 전력소모가 큰 비중을 차지한다고 볼 수 있다. 글리치 발생을 제거하는 방법으로 구조레벨에서는 조합논리회로의 모든 경로에 대해서 지연시간을 맞추는 방법이나^[2], 조합논리회로 사이에 레지스터를 삽입하여 논리깊이를 줄이는 방법^[3] 등이 제시되었다. 참조 문헌^[20]에서는 글리치 제거기(glitch canceller)를 제안하여 MAC(Multiplier And Accumulator)에 적용하였으나, 글리치 제거로 인한 전력감축보다는 글리치 제거기 자체의 전력증가가 더 큰 것이 단점으로 지적된다. 문헌^[21]에서는 비임계 경로에 위치하면서 출력 단의 글리칭 액티비티가 높고 부하 캐패시턴스 값이 큰 표준 CMOS셀을 F-게이트(Frozen gate)로 대체하여 글리칭

전력소모를 줄이고자 하였으나, F-gate는 출력 단에 발생하는 1-0로 천이 되는 글리치는 제거되나, 0-1로 천이 되는 글리치는 필터링 하지 못하여, 전력감축 효과가 낮은 단점이 있다. 데이터 패스 및 제어기에서 발생하는 글리치를 제거하는 다양한 기법 및 실험결과는 [13]에 잘 정리되어 있다.

II. 기존 ALU의 구조

그림 1은 마이크로 프로세서 및 DSP 시스템에서 사용되고 있는 16비트 기존 ALU의 블록 다이어그램을 나타낸다^[4]. 이 구조는 고정 소수점 및 부호연산을 수행하며 16개의 명령어 집합을 갖는다. 수행하는 명령어 집합은 ADSP 2101의 ALU 명령어 집합과 동일하며^[23], 각 명령어에 따른 P/G 블록의 출력 값은 문헌 [7]과 같다. 또한 트랜지스터 레벨에 의한 설계이므로 P/G블록을 프로그램 가능한 NMOS 패스 트랜지스터 회로^[5-8]로 구현할 수 있어, 게이트 레벨 설계에 비하여 하드웨어를 최소화 할 수 있는 장점이 있다^[12]. ALU 제어기는 연산코드에 따라서 P/G블록의 제어신호를 발생시킨다^[7-8,11,12]. 일반적인 RISC 및 DSP시스템에서 명령어를 해독하는 주제어기(main controller) 이외에 각 연산유닛인 ALU, MAC, 쉬프트 등의 연산블록을 제어하는 서브 제어기를 필요로 하며, 그림 1의 ALU제어기가 서브 제어기로 동작한다^[22]. 이와 같이 서브 제어기가 필요한 이유는, 각 연산블록의 독립성을 증가시키고, 주 제어기에서의 제어신호의 수를 줄임으로써 임계경로에 의한 시스템의 성능저하를 막기 위함이다^[22]. AD신호는 현재의 연산코드가 산술 또는 논리연산인지를 감지하는 신호로써, 산술연산시에는 논리 '1'이 되고 논리연산시에는 논리 '0'가 된다^[12]. ALU는 연산결과이외에 여

러 개의 플레그신호(일명 상태신호)를 발생시키는데, 이중에서 제로 플레그는 ALU의 모든 비트를 검사해야 하므로 이 경로가 ALU의 임계경로를 형성한다^[11].

일반적으로 글리치는 조합논리회로에서 여러 개의 입력신호사이에 도착시간의 차이에 의해서 발생된다^[13-15]. ALU에 사용되는 일반적인 P/G블록의 회로도는 그림 2에서 보듯이 4비트의 데이터 신호와 4비트의 제어신호가 있다. P/G블록의 데이터 신호(X, Y)는 입력 레지스터로부터 직접 도착하는데 비하여, 제어신호(L[0], L[1], L[2], L[3])는 ALU연산코드가 ALU 제어기를 통하여 해독된 이후에 P/G블록의 제어신호로 도착한다. 즉, 그림 1과 같은 파이프라인 구조를 갖는 ALU에서 데이터 신호와 연산코드는 레지스터로부터 동일한 시점에서 입력되나, P/G블록으로 입력되는 데이터 신호와 제어신호사이에는 ALU 제어기의 지연시간만큼의 도착시간 차이가 발생하게 된다. 이로 인하여 P 및 G블록의 출력 단에 많은 글리치가 발생된다. 글리치 발생을 분석적으로 설명하기 위하여, 기존의 P/G블록의 회로는 그림 2를 가지고 설명하기로 한다.

P/G블록의 출력신호는 임의의 순간에 데이터 신호에 따라서, 입출력이 연결되는 4개의 경로 중에서 하나의 경로가 선택되어 출력 값이 결정된다. 한편, 4비트의 제어신호사이에는 ALU 제어기에서 통과하는 로직레벨이 다르기 때문에, P/G블록에 도착하는 시간차이가 발생한다. 또한 데이터 신호와 제어신호사이에는 ALU 제어기의 전파지연시간 만큼 P/G블록에 도착하는 시간차이가 발생한다. 이때, 클럭신호에 의해서 새로운 데이터 및 제어신호가 갱신될 때, 위와 같은 도착시간 차이로 인하여 현재의 데이터 값이 이전의 제어신호에 의하여 순간적으로 계산되어지는 과정에서 글리치가 발생한다. 즉, P/G블록의 입출력을 연결하는 4개의 경로 중에서 새로운 데이터신호에 의해 액티브(active)되는 경로가 바뀌고, 그 경로에 해당되는 제어신호가 현재의 연산코드에 의해 발생하는 제어신호의 값이 변하는 경우에 글리치가 발생한다.

이렇게 발생한 P/G블록의 글리치들은 다음 단 기능블록으로 전파되면서 글리칭 전력소모가 증가된다. 최근 연구에 의하면, 임의의 회로에서 발생한 글리치들이 데이터패스 블록으로 전파되는 경우에 더 많은 천이를 발생시킴으로써 전력소모에 큰 영향을 미친다고 보고되었다^[13-15]. 따라서 P/G블록을 갖는 ALU의 저전력 실현을 위해서는 1차로 P/G블록의 출력 단에 글리치 발

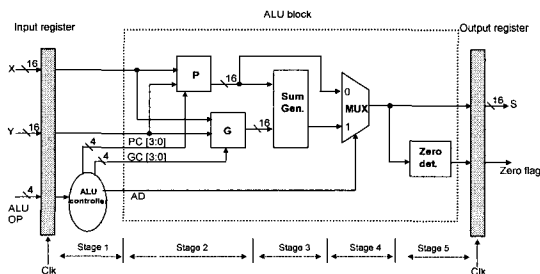


그림 1. 기존의 16비트 ALU의 블록 다이어그램.
Fig. 1. Block diagram of a conventional 16-bits ALU.

생을 제거하고, 이렇게 함으로써 발생된 글리치가 다음 단으로 전파되는 것을 막아야 한다.

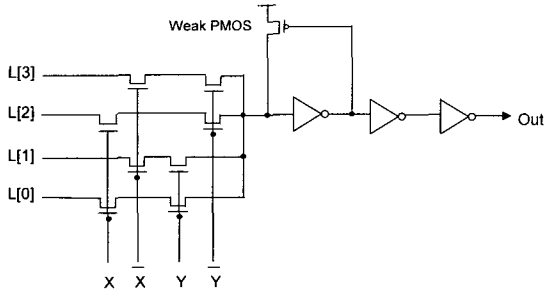


그림 2. 기존의 P/G 블록의 회로도^[5-8]
 Fig. 2. Circuit diagram of conventional P/G blocks^[5-8].

III. 제안한 ALU의 구조

앞 절에서 설명한 바와 같이 기존의 ALU구조에서는 P/G블록에 입력되는 데이터 신호와 제어신호사이의 도착시간 차이 때문에, P/G블록의 출력 단에 글리치가 발생되고 이렇게 발생된 글리치들이 다음 단으로 전파하는 단점이 있다. 이러한 단점을 해결하기 위하여 그림 3과 같은 구조를 제안하였다.

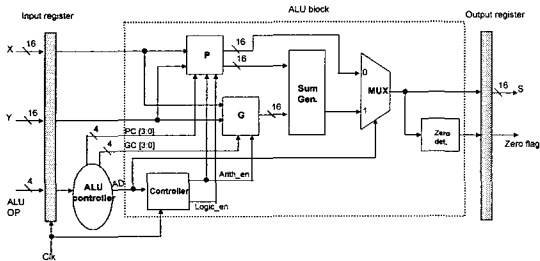


그림 3. 글리치 전파를 제거한 제안한 16비트 ALU의 블록 다이어그램
 Fig. 3. Block diagram of a proposed 16-bits ALU to remove glitch-propagation.

그림 3은 [12]에서 제안한 저전력 방법인, 논리연산시의 Sum 블록의 불필요한 스위칭 억제 방법을 채용하면서 본 논문에서 제기한 글리치 발생 및 전파를 제거한 구조이다. P블록의 출력은 논리연산이 수행되는 경우에 Sum 블록의 불필요한 스위칭 억제를 위해, 산술연산용 출력과 논리연산 출력용으로 분리된다. 저전력 실현을 위한 논리연산시의 Sum 블록의 불필요한 스위

칭 억제방법은 실제로 ALU의 주된 연산이 산술연산임을 감안할 때 전력감축효과가 크지 않다^[12]. 그림 3에서는 P/G블록내에 래치를 삽입하여 P/G블록에 입력되는 모든 신호가 도착되었을 때 래치의 인에이블 신호를 액티브 시킴으로써, P/G블록의 출력 단에 글리치를 제거하였다. 래치의 인에이블 신호는 제어기로부터 발생되며, 이 신호는 P/G블록의 출력 단에 글리치가 발생하지 않도록 출력신호를 인에이블시키는 제어신호이다. Arith_en신호는 산술연산이 수행되는 경우에 P/G블록의 출력 단의 글리치를 제거하고, Logic_en신호는 논리연산이 수행되는 경우에 P블록의 출력 단의 글리치를 제거시킨다. 제거의 자세한 논리회로도 및 타이밍도는 다음 2항에서 설명된다. 제안한 구조는 기존 구조와 비교하여 다음과 같은 두 가지 관점에서 글리칭 전력소모를 최소화시킨다. 첫째로는 P/G블록의 출력신호에 글리치가 발생하지 않기 때문에, 다음 단 기능블록으로의 글리치가 전파되지 않는다. 둘째로는 Arith_en 신호를 가지고 P/G블록의 출력신호를 동시에 인에이블 시키기 때문에, 다음 단 블록인 Sum블록으로 인가되는 두 신호를 동기화(synchronization)시킴으로써 Sum블록의 첫 단에서 글리치 발생 가능성을 제거시킨다.

1. 글리치 전파차단을 위한 P/G블록의 회로

조합논리회로의 입력신호사이의 도착시간차이로 인한 글리치 발생을 제거하기 위한 일반적인 방법은 ‘지연소자의 삽입’으로써, 이것은 가장 늦은 신호에 기준을 두고 그 외의 신호들에 대해서 지연소자를 삽입하여 모든 신호에 대해서 지연시간을 균등하게 하는 것이다^[13,21]. 이러한 ‘지연소자의 삽입방법’은 일반적으로 회로의 입력수가 비교적 적은 경우에 유용하며, P/G블록과 같이 다수의 입력 수를 갖는 회로에 대해서는 회로 복잡도 증가 및 공정변화 등에 영향을 받는 입력 수가 커지기 때문에 효과적이지 않다. 다수의 입력 수를 갖는 회로의 글리치를 제거하는 방법으로는 제어회로를 추가하는 방법이 효과적이며, 이에 대한 전형적인 예가 ‘래치삽입방법’이다^[15]. 이것은 글리치가 발생하는 노드 상에 인에이블 신호를 갖는 래치를 삽입하는 방법으로 다음과 같은 단점을 가진다. 첫째로는, 래치삽입에 따른 회로의 과도한 지연시간의 증가이다^[16]. 저전력 회로설계의 기본목표는 기존회로의 성능을 유지하면서 회로의 전력소모를 줄이는 것인데^[1-3], 저전력을 실현하면서 그에 따른 과도한 지연시간 증가는 바람직하지 않은

결과이다. 둘째로 래치의 인에이블 신호의 타이밍은 최적의 회로성능(최소의 지연시간 증가 및 완벽한 글리치 제거)을 위해서 단 하나의 인에이블 시점이 요구된다. 즉, 글리치 제거를 위한 인에이블 신호의 타이밍은, 글리치 신호를 갖는 데이터 신호가 완전히 안정화된(stabilized) 시점에서 결정된다. 예를 들면, 인에이블 신호의 타이밍이 데이터 신호가 안정화되기 전에 이루어지면 출력 단에 글리치가 완벽하게 제거되지 않으며, 반대로 인에이블 신호의 타이밍이 데이터의 신호가 안정화가 되고 나서 한참 후에 이루어지면 글리치는 완벽하게 제거되나 래치에 의한 지연시간 이외에, 부가의 지연시간이 증가되는 문제가 발생한다. 한편, 모든 전자 회로의 지연시간은 공정, 전원전압, 온도 등과 같은 환경변화의 함수이므로, 모의실험상에서 인에이블 신호의 타이밍을 최적화시켜도 칩 제조 후에 공정변화로 인한 지연시간의 변화때문에 최적의 회로성능을 기대하기 어렵다. 상기에 기술한 두 가지 문제점으로 인하여 실제로 다수의 입력 수를 갖는 회로에 대해서는 래치를 삽입하는 방법은 효과적이지 않다. 그런데 본 논문의 제안한 P/G블록안의 래치삽입에 따른 문제점들은, 기존의 P/G블록의 회로구조 때문에 완화되거나 또는 해결된다. 이것은 기존의 P/G블록의 회로구조가 이미 래치와 유사한 형태를 띤 특수한 구조를 가지기 때문이다. 글리치 제거를 위해 사용되는 래치삽입은 기존의 회로에 래치를 삽입해야 한다. 따라서 본 논문에서 P/G블록의 글리치 제거를 위해 래치를 삽입해야 하는데, 이미 기존의 P/G 회로가^[5-8] weak PMOS와 인버터가 피드백으로 연결되어 있어 래치가 유사한 형태로 있다. 즉, 특수한 회로구조란 래치삽입의 대상회로의 일부분이 래치와 유사한 회로형태를 취하고 있는 경우를 말한다.

그림 4와 그림 5는 제안한 ALU에 사용된 P/G블록의 회로도들을 나타낸다. 글리치 제거를 위해 P/G블록안에 인에이블 신호를 갖는 래치를 삽입하였으며, 2개의 인에이블 신호로는 Arith_en 및 Logic_en 신호가 사용된다. 이 2개의 인에이블 신호는 제어기로부터 발생된다. 래치의 인에이블 신호는 P/G블록에 인가되는 입력경로의 신호가 도착한 이후에 액티브됨으로써, P/G블록의 출력 단에 글리치를 제거시키는 역할을 한다. 2개의 인에이블 신호가 각각 필요한 이유는, ALU의 모든 연산에 대해서 글리치를 제거하기 위해서이다. 즉, 산술연산이 수행되는 경우에는 Arith_en 신호만 액티브되고, 논

리연산이 수행되는 경우에는 Logic_en 신호만 액티브된다. 이 회로에서 가장 중요한 것은 Arith_en 신호와 Logic_en 신호의 타이밍이다. 두 개의 인에이블 신호의 역할은, 첫째 P/G블록의 출력 단의 글리치 발생 및 다음 단으로의 글리치 전파를 제거하며 둘째로는 논리연산의 Sum 블록에서의 불필요한 스위칭을 억제하는 역할을 한다^[12].

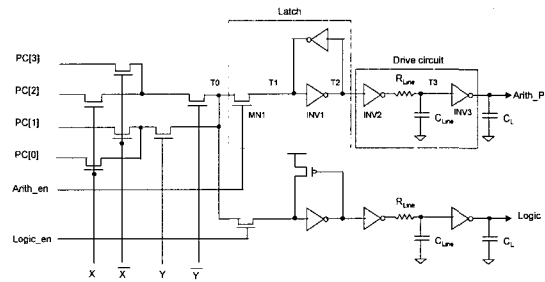


그림 4. P블록의 회로도
Fig. 4. Circuit diagram of a P block.

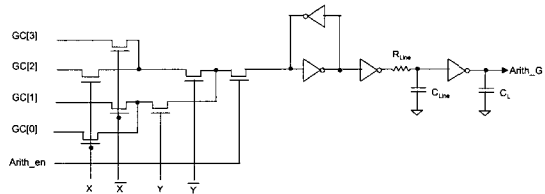


그림 5. G블록의 회로도
Fig. 5. Circuit diagram of a G block.

먼저 래치삽입에 따른 과도한 지연시간 증가의 문제점에 대해서 언급하기로 한다. 기존의 P/G블록은 출력단의 구조가 정적 래치와 유사한 구조를 가진다^[5-8,11-12]. 그림 2에서 P/G블록의 출력 단은 weak PMOS와 인버터가 서로 연결된 구조로써, 이 모양은 정적 래치의 구조와 유사하다. 즉, 기존 P/G블록에 래치 삽입을 위한 추가의 회로로는 NMOS 패스 트랜지스터와 페드백 인버터에 사용되는 NMOS 하나가 필요하며, 실제의 지연시간 증가에 영향을 주는 것은 NMOS 패스 트랜지스터뿐이다(그림 4의 MN1). 따라서 제안한 P/G블록이 기존 회로와 비교하여 NMOS 패스 트랜지스터 하나만큼의 지연시간 증가가 있다.

두 번째 문제점으로 제기된 인에이블 신호의 타이밍은, 그림 4 및 그림 5에서 래치의 출력이 곧바로 Sum 블록에 연결되지 않고 구동회로를 거쳐서 Sum 블록으로 입력된다. 그림 4에서 구동회로는 전력소모를 최소

화시키기 위해서 삽입하였다. 그림 4에서 T_0 노드는 NMOS 패스 트랜지스터에 의해서 계산되므로, 논리 '1'에 대응되는 전압레벨이 NMOS의 문턱전압만큼 낮아진다. 논리 '1'에 대응되는 전압레벨이 작아짐에 따라서, 다음 2-1과 같은 식에 의하여 T_2 노드의 상승시간이 증가하게 된다^[8]

$$t_r = k \frac{C_L}{\beta V_{DD}} \quad (2-1)$$

여기서 C_L 은 부하 캐패시턴스이고, β 는 트랜지스터의 이득값이다. 이렇게 NMOS의 문턱전압만큼 낮아진 논리 '1'의 전압레벨은 피드백 인버터에 의해서 전원전압레벨로 복원된다. 피드백 인버터에 의하여 T_1 노드의 전압이 전원전압레벨로 복원되기 전까지는, INV1의 NMOS는 강하게 도통되어 있고 PMOS는 약하게 도통되어 있기 때문에 INV1에서 정적전류가 발생된다. 한편, 구동회로가 없는 경우에 INV1의 부하 캐패시턴스 값은 T_2 노드의 팬아웃이 크므로 이에 따라서 부하 캐패시턴스 값이 커지게 된다. 이러한 두 가지 상황에서, INV1의 트랜지스터 크기에 따라서 회로성능변화를 고찰해 보면 다음과 같다. 먼저, INV1의 크기를 작게 하면 출력노드의 부하 캐패시턴스 값이 크기 때문에 INV1의 지연시간이 증가하게 된다. 이것은 전체 ALU의 지연시간이 증가될 뿐만 아니라, 피드백 인버터를 늦게 동작시킴으로써 INV1에서의 정적전류량이 커진다. 이러한 문제점 해결을 위해 INV1의 트랜지스터 크기를 증가시키면 되나, 이 경우에는 INV1의 단락회로 전류가 커지게 된다. 부하 캐패시턴스가 없는 경우에 CMOS 인버터의 단락회로 전류는 다음 2-2와 같은 식으로 표현된다^[8].

$$P_{sc} = \frac{\beta}{12} (V_{DD} - 2V_t)^3 \frac{t_f}{t_p} \quad (2-2)$$

여기서 V_t 는 MOSFET의 문턱전압이고, t_f 는 입력신호의 상승/하강시간이며, t_p 는 입력신호의 주기를 나타낸다. 이때 T_1 노드의 t_r 는 전술한 바와 같이 큰 값을 가지며, INV1의 트랜지스터 크기가 커짐에 따라서 β 값이 커지게 된다. t_f 및 β 가 모두 큰 값을 가지기 때문에 식 (2-2)에 의해서 INV1의 단락회로 전류가 증가하게 된다. 따라서 INV1의 트랜지스터 크기에 따라서 INV1에서 발생하는 단락회로 전류와 정적전류를 동

시에 줄이기 위해서는, INV1의 크기를 가능한 작게 하면서 출력단의 부하 캐패시턴스를 줄여야 한다. 이를 위해서 INV1의 크기를 최소로 하고, 다음 단계 최소 크기의 인버터(INV2)를 삽입하였다. INV3는 팬아웃이 크기에 따라서 큰 부하 캐패시턴스 값을 가지므로 적당한 크기로 증가시켜야 한다.

한편, 래치 뒤의 구동회로 때문에 그림 4에서 T_1 노드는 글리치를 완벽하게 제거할 필요가 없으며, 다만 P/G블록의 최종출력에만 글리치가 발생되지 않으면 된다. 이것은 저전력 실현을 위한 가장 큰 목표가 다음 단 블록으로 글리치 전파를 차단하는데 있기 때문이다. T_1 노드의 신호가 작은 펄스 폭을 갖는 글리치가 발생되더라도 이 신호가 뒷 단계의 구동회로를 거치면서 최종 출력에서는 글리치가 제거되는데 이것을 글리치 필터링^[217] 또는 글리치 흡수(glitch absorption)^[18]라고 한다. 즉, 작은 펄스 폭을 갖는 글리치들은 다음 단계 회로를 통과하면서 자동적으로 제거된다. 일반적으로 임의의 게이트 출력에서 발생한 글리치는 반드시 다음 두 가지 경우중의 어느 하나로 나타난다^[17,18]. 한번 발생한 글리치는 글리치의 폭과 다음 단계 회로의 전파지연시간에 따라서 다음 단계 회로로 전파되든지 또는 다음 단계 회로에 의해서 필터링 된다^[18]. 글리치가 다음 단계 회로에 의해서 필터링 되는 구체적인 이유는 다음과 같다. 발생한 글리치가 다음 단계 회로인 CMOS 인버터로 전파되는 경우를 가정하자. 일반적으로 CMOS 인버터는 잠음여유의 극대화를 위해 논리문턱전압을 $1/2V_{dd}$ 에 설정한다. 이것은 인버터의 입력전압의 스윙이 $1/2V_{dd}$ 이상 변했을 경우에는 출력이 천이 하지만, 입력전압의 스윙이 $1/2V_{dd}$ 보다 작은 경우에는 인버터의 출력은 천이하지 않는다. 따라서 CMOS 인버터로 입력되는 글리치의 진폭이 $1/2V_{dd}$ 보다 작은 경우에는 인버터의 출력이 천이하지 않으며, 이것으로 인해 글리치가 CMOS 인버터에 의해 필터링 된다.

그림 6은 본 논문에서 나타나는 글리치 필터링 현상 및 Arith_en 신호의 지연시간 변화에 대한 둔감함을 보여주는 그림이다. 본 논문에서 나타나는 글리치 필터링 현상을 그림 4와 그림 6을 가지고 자세히 설명하기로 한다. 데이터신호와 제어신호와의 도착시간의 차이로 인해 발생하는 큰 펄스폭(W_1)을 갖는 글리치가 발생하는 T_0 노드의 신호는, t_{min} 지점에서 인에이블 되는 Arith_en신호에 의하여 T_1 노드에서는 글리치의 폭(W_2)이 감소된다. 이렇게 감소된 폭을 갖는 글리치가 INV1

을 통과하면서 글리치의 폭이 더욱 감소되고, 이것이 다시 INV2를 거치면서 글리치의 진폭이 $1/2V_{dd}$ 보다 작게 된다. 따라서 글리치의 진폭이 $1/2V_{dd}$ 보다 작은 신호가 INV3를 통과하면서 최종 출력노드인 'Arith_P'에서는 글리치가 완전히 제거된다. 글리치 필터링에 의하여 최종출력에 글리치가 제거되는 조건에서, 허용되는 Arith_en신호의 타이밍의 최소값(t_{min}) 및 최대값(t_{max})을 표시하였다. 최대값은 T₀노드의 신호가 안정화되는 시점에서 결정되며, 최소값은 회로 시뮬레이션을 통하여 결정하였다. P/G블록의 최종출력에 글리치 발생을 제거하기 위한 Arith_en 신호의 타이밍은 t_{min} 에서 t_{max} 사이의 시간구간(Δt)이 존재한다. 따라서 래치의 인에이블 신호의 타이밍이 한 시점에서 결정되는 것이 아니라, 유한한 시간구간이 존재하며 이것은 지연시간 변화에 대한 둔감함을 나타낸다.

2. Controller

그림 7은 제안한 구조에 사용된 제어기의 논리회로도를 나타낸다. 두 개의 출력신호 중에서 Arith_en신호는 산술연산시에 P/G블록의 출력을 인에이블시키는 역할을 하며, Logic_en신호는 논리연산시에 P블록을 인에이블시키는 역할을 한다. 이와 같이 두 개의 인에이블 신호가 각각 필요한 이유는, ALU의 모든 연산(산술/논리연산)에 대해서 다음 단으로의 글리치 전파를 제거하기 위해서이다. 지연소자는 인버터 체인이 사용되었고, 인버터 체인은 ALU 제어기의 지연시간 및 P/G블록의 임계경로의 지연시간에 의해서 결정된다. 그림 8은 제어기의 자세한 타이밍도를 나타낸다. 그림에서 보듯이, 두 개의 인에이블 신호가 주어진 ALU연산에 대해서 하나의 신호만 인에이블됨을 알 수 있다.

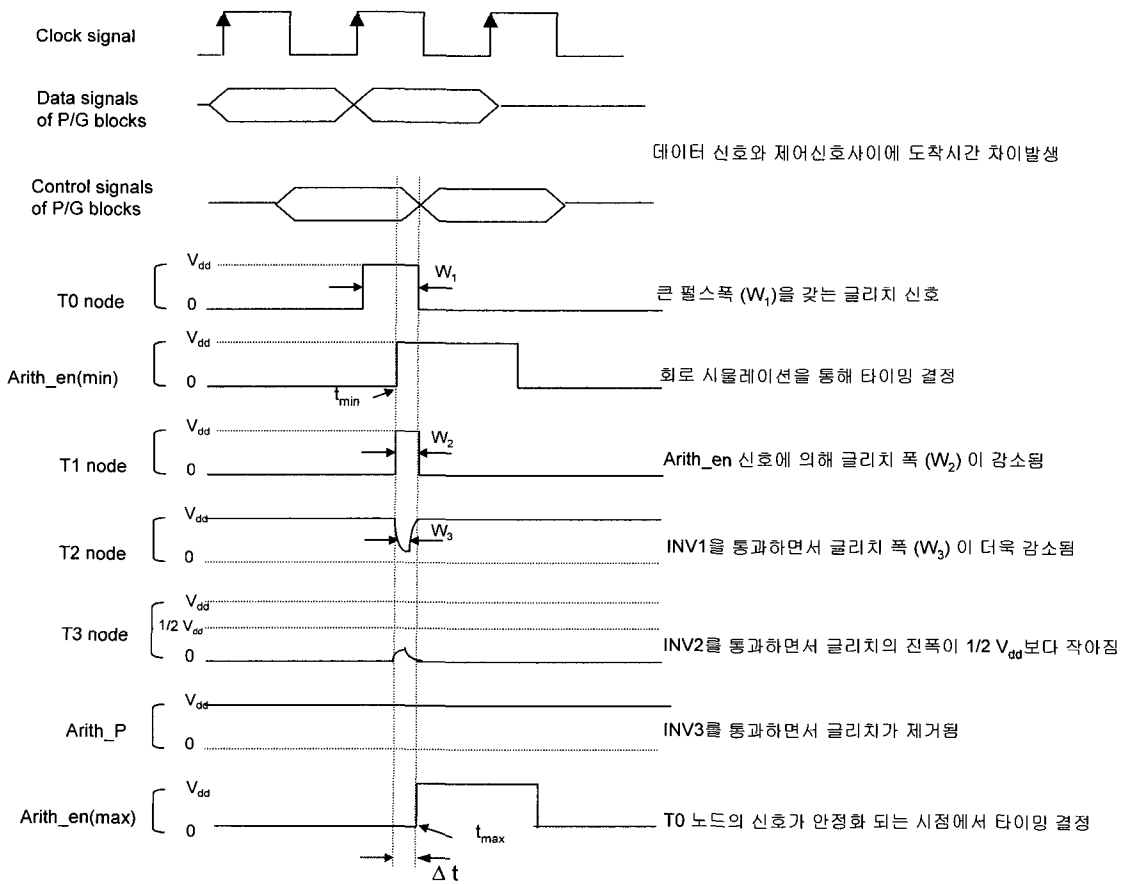


그림 6. 글리치 필터링 및 Arith_en신호의 지연시간 변화에 대한 둔감성
 Fig. 6. Glitch filtering and Insensitivity of an Arith_en signal to delay time variation.

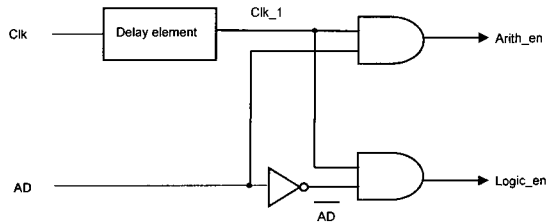


그림 7. 2개의 인에이블 신호를 발생시키는 제어기의 논리 회로도

Fig. 7. Logic diagram of a controller to generate two enable signals.

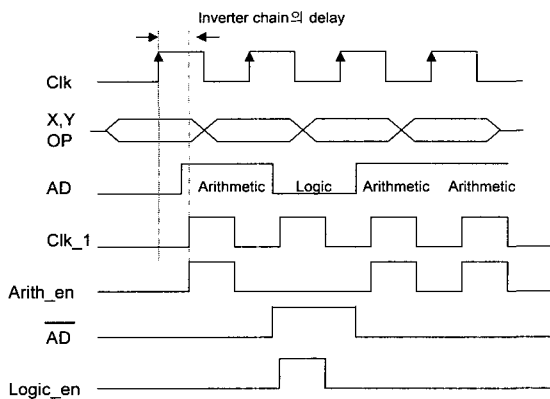


그림 8. 제어기의 자세한 타이밍도

Fig. 8. Detailed timing diagram of a controller.

IV. 실험결과

0.6 μ m 단일폴리, 삼중금속 CMOS 공정변수를 가지고 16비트 ALU를 그림 1 및 3과 같은 기존 구조 및 제안한 구조를 각각 구현하여 HSPICE를 이용하여 모의실험을 하였다. 동작전압은 3.3V, 온도는 25 $^{\circ}$ C, 입력신호의 상승/하강시간은 1ns, 입력신호의 주파수는 100MHz, 표준공정의 모델변수를 사용하였다. P/G블록은 앞서 언급한 바와 같이, 기존구조와 제안한 구조에서 서로 다른 회로로 구현하였고, Sum 블록은 여러 가지 병렬 덧셈기중에서 PDP(Power Delay Product)면에서 가장 효율적인 ELM 덧셈기로 기존 및 제안한 구조를 동일하게 구현하였다^{[24]-[25]}.

1. 전력소모

그림 9는 랜덤 입력패턴이 인가된 경우에 HSPICE의 모의실험 결과 파형을 나타낸 그림으로써, 그림 하단에는 기존구조의 P블록의 상위 3비트 신호(P[15], P[14], P[13])의 글리치 발생모습과, 그림 상단에는 이에 대응되는 제안한 구조의 P블록의 상위 3비트 신호의 글리치 제거모습을 보이고 있다. 또한 클럭신호와 AD신호로부터 발생되는 두 개의 인에이블 신호인 Arith_en 및 Logic_en 신호의 파형도 볼 수 있다.

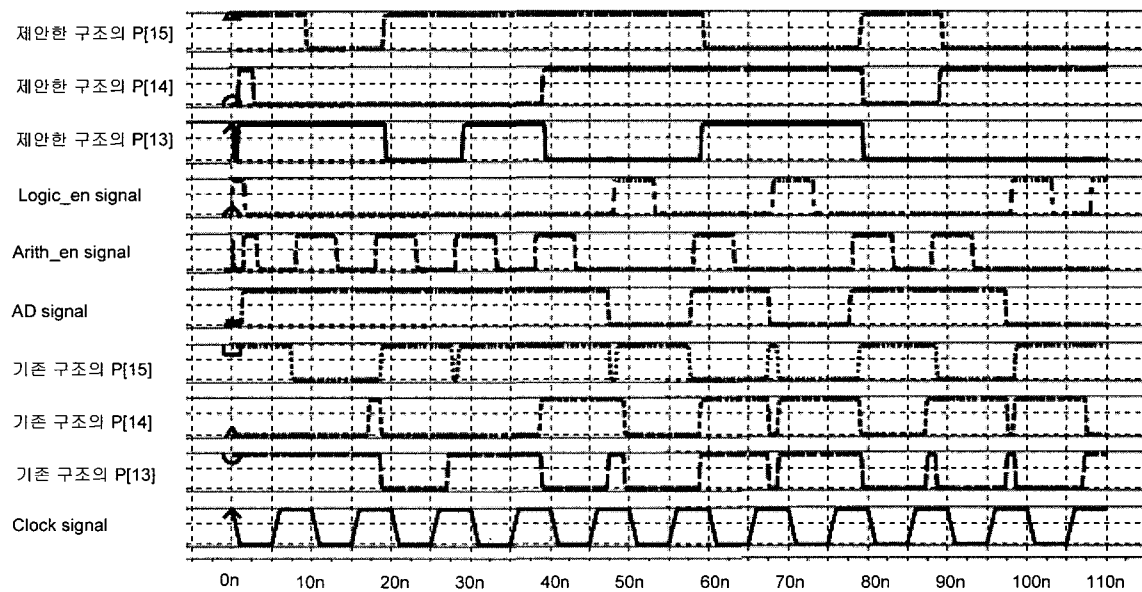


그림 9. 모의실험 파형

Fig. 9. Simulation waveform.

본 논문에서는 제안한 아이디어의 검증을 위하여 ALU 전체의 평균전력소모 및 각 블록에 대한 전력소모를 모의실험 하였다. 평균전력소모는 산술 대 논리연산의 비가 7 : 3일 때, 기존 구조와 제안한 구조의 각 블록 및 전체 ALU의 전력소모를 비교하였다. ALU 연산 중에서 산술 대 논리연산의 비율은 실제 ALU를 내장한 DSP 및 RISC 프로세서의 응용분야에 따라서 크게 차이가 난다. 표 1은 인텔 80×86 마이크로 프로세서로 5개의 SPECint92 프로그램을 실행시켰을 때, 각 명령어에 대한 빈도수를 나타낸 표이다^[26]. 표에서 보듯이 ALU 명령어가 전체 명령어 중에서 37%를 차지했으며, ALU 명령어 중에서 산술명령어가 78%, 논리명령어가 22%를 차지하였다. 산술 대 논리연산의 비율은 DSP 및 RISC 프로세서의 응용분야에 따라서 크게 차이가 나지만, [26]-[28]을 토대로 평균전력소모를 위한 산술 대 논리연산의 비율을 7 대 3으로 설정하였다.

표 2는 평균전력소모에 대한 실험결과를 나타낸 표이다. 표에서 보듯이 Sum 블록 및 P, G블록이 기존 구조와 비교하여 각각 55.7%, 51.2%의 전력감축을 나타내었다. P/G블록에서의 전력감축의 원인은 글리치 발생

제거로 인한 것이며, Sum 블록에서의 전력감축은 산술/논리연산시에는 글리치 전파제거 및 논리연산시의 Sum 블록의 불필요한 스위칭 억제가 그 원인이 된다. 한편 ALU블록내의 Sum 및 P/G블록이외의 다른 블록(others)에서 제안한 구조가 제어기가 추가되었음에도 불구하고 전력소모가 약간 증가한 이유는, 제어기의 추가로 인한 전력증가보다는 Logic_en신호의 작용으로 MUX로의 글리치 전파제거로 인하여 MUX블록에서의 전력감축이 더 컸기 때문이다.

표 3은 RISC 및 DSP의 응용분야에 따라서 수행되는 ALU의 산술 대 논리연산의 비율이 달라지는 경우에 대하여, 기존 및 제안한 ALU의 전력소모 및 전력감축률을 나타낸 표이다. 표에서 보듯이 수행되는 논리연산의 비율이 산술연산에 비하여 증가할수록 제안한 ALU의 전력감축률이 커지는 이유는, 제안한 ALU가 논리연산시의 Sum블록에서의 불필요한 동적 전력소모를 제거시켰기 때문이다^[12].

한편 표 2에 제시된 제안한 ALU 블록의 42.2%의 전력감축률은, 실험조건이 산술 대 논리연산의 비율을 7 대 3으로 설정하여 수행된 것이므로 다음 두 가지의

표 1. 5개의 SPECint92 프로그램에 대한 80×86 프로세서의 명령어 비율^[26]
Table 1. 80×86 instruction percentage for five SPECint92 programs^[26].

80×86 응용프로그램	로드/스토어 명령어	분기 명령어	기타 명령어 (move, call)	ALU		합계
				산술연산	논리연산	
SPECint92	34%	21%	8%	37%		100%
				78%	22%	

표 2. 산술 대 논리연산 비율이 7 대 3일 때의 기존구조와 제안한 16비트 ALU와의 평균전력소모 비교
Table 2. Comparison of average power consumption between conventional and proposed 16-b ALU with the activity ratio of arithmetic to logic by 7 to 3.

블록	서브 블록	기존 구조(mW)	제안한 구조(mW)	전력감축률(%)
ALU	Sum 블록	8.87	3.93	55.7%
	P/G 블록	5.08	2.48	51.2%
	기타	4.94	4.50	-
	ALU 블록	18.89	10.91	42.2%
입출력 레지스터		4.72	4.52	-
ALU 제어기		1.48	1.49	-
클럭 회로망		2.59	2.60	-
전체 전력소모		27.68	19.52	29.5%

※ 기타 = MUX, 제로 감지기, 제어기(제안한 구조) 및 각종 버퍼 등을 포함

요인에 의해서 나타난 결과이다. 그 첫 번째는 본 논문에서 제시한 글리치 제거의 의한 효과이고 두 번째는 제시한 논리연산시의 Sum 블록의 불필요한 스위칭을 억제한 효과이다^[12]. 따라서 전체 전력감축 중에서 글리치 전파제거에 의한 효과와, 논리연산시의 Sum 블록의 불필요한 스위칭 억제 효과를 나눌 필요가 있다. 글리치 제거에 의한 전력감축효과만을 알아보기 위해서 모의실험조건을 산술연산만 수행시켜, 기존구조와 제안한 구조의 평균전력소모를 비교하였다. 실험결과로부터 글리치 발생 및 전파제거로 인하여 28.3%의 전력감축이 있음을 알 수 있다. 따라서 표 2에 나타난 42.2%의 전체 전력감축률은 글리칭 전력감축에 의한 28.3%와 논리연산시의 Sum 블록의 스위칭 억제에 의한 13.7%로 나누어 분석할 수 있다.

2. 지연시간

ALU의 지연시간 실험을 위해서는 보다 세밀한 분석이 선행되어야 한다. 그것은 ALU를 구성하는 각각의 블록(ALU 제어기, P/G블록, Sum 블록 및 제로 감지기)에 대한 최악 입력조건이 각각 존재하나, 문제는 ALU전체 시스템에서 하나의 입력벡터가 4개의 블록에

대해 모두 최악조건을 항상 만족하지 않는다. 따라서, ALU의 지연시간 계산을 위해서는 4개의 블록들의 임계경로가 동시에 계산되는 입력벡터가 인가되어야 한다. 그러한 입력벡터를 가지고 16비트 ALU의 지연시간에 대한 실험결과는 표 4에 나타나 있다. 제안한 ALU에서 0.1ns만큼의 지연시간 증가의 원인은, 제안한 구조에 사용된 P/G블록의 NMOS 패스 트랜지스터의 삽입에 의한 것이다.

3. 공정변화에 대비한 clk-to-Arith_en 신호의 지연시간구간 결정

앞 절에서 서술한 바와 같이, 제안한 구조의 핵심은 Arith_en신호의 타이밍 발생이다. 이를 위하여 ALU제어기의 7비트의 출력신호에 대한 지연시간을 모의실험하였으며, 이 결과를 표 5에 요약하였다.

그림 4에서 T0노드의 신호가 안정화되는 시점은 ALU 제어기의 임계경로의 지연시간과 제어신호가 NMOS 스위치까지 전파하는 시간의 합으로 주어지며, Arith_en 신호의 t_{max} 는 표 5로부터 2.3ns를 얻었다. t_{min} 은 P/G블록의 최종 출력 단에 글리치가 나타나지 않는 조건하에서 모의실험을 통하여 t_{max} 값을 감소시켜가면

표 3. 산술 대 논리연산 비율에 따른 기존구조 및 제안한 16비트 ALU의 평균전력소모 비교

Table 3. Comparison of average power consumption of both conventional and proposed 16-b ALU according to the activity ratio of arithmetic to logic operation.

산술 대 논리연산 비율	기존 구조(mW)	제안한 구조(mW)	평균전력소모 감축률(%)	PDP 감축률(%)
10 : 0	16.41	11.76	28.3	27.4
9 : 1	17.94	12.58	29.9	28.9
7 : 3	18.89	10.91	42.2	41.4
5 : 5	15.86	8.22	48.2	47.5
3 : 7	16.39	7.06	56.9	56.3
1 : 9	15.51	5.24	66.2	65.8
0 : 10	14.93	4.56	69.5	69.1

표 4. 기존 및 제안한 16비트 ALU의 지연시간 비교

Table 4. Comparison of delay time between 16-b conventional and proposed ALU.

	지연시간(ns)				
	ALU 제어기	P 블록	Sum 블록 (MUX)	제로 감지기	합계
기존 구조	2.10	1.30	2.78	1.03	7.20
제안한 구조	2.10	1.40	2.78	1.03	7.30

서 최소값을 구하였다. 실험결과, $t_{min}=1.7ns$ 및 $t_{max}=2.3ns$ 값을 얻었으며 모의실험을 통하여 이 시간구간에서는 P/G블록에서 클리치가 완벽히 제거됨을 확인하였다. 표 5를 토대로 하여 clk-to-Arith_en신호의 지연시간 변화에 따른 제안한 회로의 성능변화를 모의실험 하였으며, 그림 10에서 그림 12가 이에 대한 결과를 나타낸다. 이때 가로축의 인에이블 신호의 지연시간 변화량은 제안한 ALU에서 제어기(controller)내에 사용된 인버터 체인의 두 단의 인버터에 대한 지연시간이다. 그림 10은 clk-to-Arith_en신호의 지연시간 변화에 따른 제안한 16비트 ALU의 지연시간을 나타낸 것으로, clk-

to-Arith_en의 지연시간이 2.3ns보다 큰 영역에서는 제안한 ALU의 지연시간이 증가하는 영역이다. 그림 11에서 clk-to-Arith_en의 지연시간이 1.7ns보다 작은 영역에서는 그림 4의 T0노드에서 발생된 클리치가 'Arith_P'노드까지 전파하며, 클리치 전파에 의한 전력소모가 큰 쪽으로 증가하는 시간영역이다. 실험결과로부터 clk-to-Arith_en의 지연시간이 1.7ns~2.3ns의 시간영역에서 비교적 평탄한 PDP 곡선을 나타내었다. 따라서 clk-to-Arith_en의 지연시간을 2.0ns에 설정함으로써, 공정변화에 대하여 약 0.3ns의 타이밍 여유가 있음을 알 수 있다.

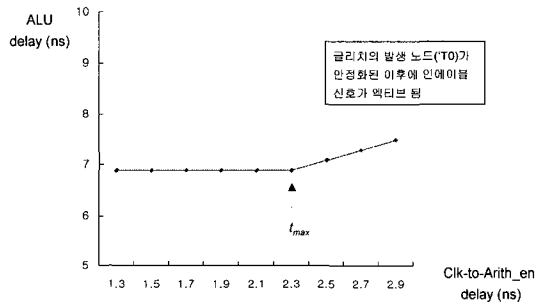


그림 10. Clk-to-Arith_en신호의 지연시간 변화에 따른 제안한 ALU의 지연시간 변화
Fig. 10. Delay time variation of the proposed ALU with clk-to-Arith_en delay variation.

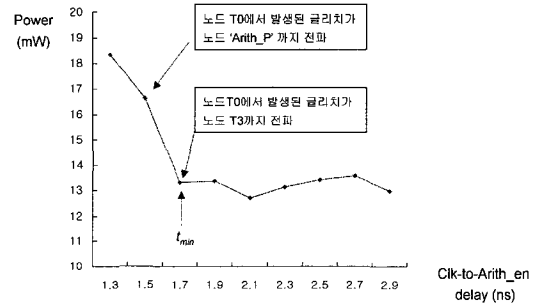


그림 11. Clk-to-Arith_en신호의 지연시간 변화에 따른 제안한 ALU의 전력소모 변화
Fig. 11. Power consumption variation of the proposed ALU with clk-to-Arith_en delay variation.

표 5. ALU 제어기의 지연시간
Table 5. Delay time of the ALU controller.

출력신호	지연시간(ns)	비고	
Clk-to-PC[3]	2.0		
Clk-to-PC[2]	2.1	입계경로	
Clk-to-PC[1]	1.9		
Clk-to-PC[0]	1.9		
Clk-to-GC[2]	1.8		
Clk-to-GC[1]	1.8		
Clk-to-GC[0]	1.8		
Clk-to-Arith	t_{min}	1.7	Clk-to-Arith_en의 지연시간이 t_{min} 보다 작으면, 클리치 전파에 의한 전력소모가 커짐
	t_{max}	2.3	Clk-to-Arith_en의 지연시간이 t_{max} 보다 크면, 제안한 구조의 지연시간이 증가함

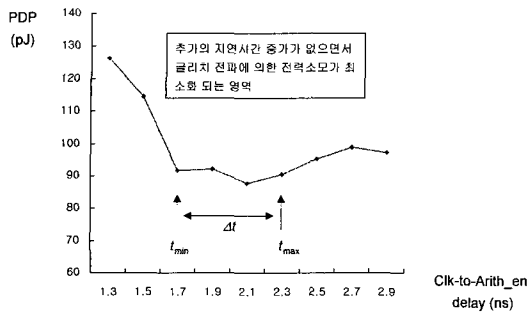


그림 12. Clk-to-Arith_en신호의 지연시간 변화에 따른 제안한 ALU의 PDP 변화

Fig. 12. PDP variation of the proposed ALU with clk-to-Arith_en delay variation.

표 6은 기존의 구조와 본 논문에서 제안한 ALU의 전체성능을 비교한 표이다. 제안한 16비트 ALU는 저전력 회로설계의 평가기준인 PDP측면에서 약 27%의 감축을 얻은 반면에, 트랜지스터 수에서는 약 5%의 증가가 있었다.

V. 결 론

본 논문에서는 기존의 P/G 블록을 갖는 ALU 구조에서 나타나는 글리칭 전력소모를 최소화시킨 새로운 ALU 구조 및 회로를 제안하고, 회로 시뮬레이션을 통하여 그 효과를 검증하였다. 기존의 ALU 구조는 P/G 블록에 인가되는 입력신호 중에서 데이터신호는 입력 레지스터로부터 직접 도착되는 반면에, 제어신호는 연산코드가 ALU 제어기를 통하여 P/G블록에 도착하기 때문에 P/G블록의 두 종류의 입력신호사이에 도착시간의 차이가 발생한다. 이로 인하여 P/G블록의 출력단에 많은 글리치가 발생하며, 이렇게 발생된 글리치들은

다음 단 기능블록으로 전파되어 글리치에 의한 전력소모로 연결된다. 제안한 구조에서는 P/G블록의 글리치 제거를 위하여 P/G블록내에 인에이블 신호를 갖는 래치를 삽입하고, 래치의 인에이블 신호는 제어기를 통하여 발생시켰다. 제안한 구조에서 최적의 회로성능(최소의 지연시간 증가와 완벽한 글리치 전파제거)을 위해서는 인에이블 신호의 타이밍이 중요하다. 글리치 제거를 위해 사용되는 래치삽입 방법은, 최적의 회로성능을 위해서 요구되는 인에이블 신호의 타이밍은 오직 하나의 시점에서 결정된다. 이러한 경우에 인에이블 신호의 지연시간은 공정변화에 민감한 단점을 가진다. 반면에 본 논문의 P/G블록내에 사용된 래치의 인에이블 신호의 타이밍은 한 시점만 존재하는 것이 아니라 유한한 타이밍 여유를 갖는다. 이것은 글리치 필터링에 의해 발생하는 현상으로, 래치의 출력신호가 직접 Sum 블록에 연결되지 않고 구동회로를 거치기 연결되기 때문이다. 이로 인하여 래치의 인에이블 신호는 공정변화에 대하여 약 0.3ns의 타이밍 여유가 있음을 모의실험을 통하여 확인하였다. 또한 기존의 P/G블록의 출력 회로가 이미 래치와 유사한 형태로 되어 있어서, 제안한 구조에 사용된 P/G블록내에 래치삽입에 따른 지연시간 증가가 최소화되었다.

0.6 μ m, 3.3V 모델 파라미터를 가지고, 16비트 ALU를 구현하여 모의 실험한 결과, 제안한 구조는 기존 구조에 비하여 28.3%의 글리칭 전력소모와 27.3%의 PDP가 각각 감소되었다.

참 고 문 헌

[1] Anantha P. Chandrakasan and Robert W. Brodersen, "Low Power Digital CMOS Design",

표 6. 산술연산만 수행되었을 때의 기존 구조와 제안한 구조의 성능비교

Table 6. Performance comparison between conventional and proposed 16-b ALU with only arithmetic operations.

성능	기존 구조	제안한 구조	증감률(%)	비고
지연시간 (ns)	7.20	7.30		
전력소모 (mW)	16.41	11.76	28.3% 감축	글리치 제거에 의한 효과만을 나타냄
트랜지스터 수	2,716	2,846	4.6% 증가	
PDP(pJ)	118.2	85.9	27.3% 감축	

- Kluwer academic publishers, 1995.
- [2] Jan M. Rabaey and Massoud Pedram, "Low Power Design Methodologies", Kluwer academic publishers, 1996.
- [3] Gary K. Yeap, "Practical Low Power Digital VLSI Design," Kluwer academic publishers, 1998.
- [4] David A Patterson and John L. Hennessy, "Computer organization & design the hardware/software interface", Morgan Kaufmann Publishers, pp. 470-476, 1998.
- [5] Peter Pirsch, "Architectures for Digital Signal Processing", John Wiley & Sons, New York, pp 306-312, 1996.
- [6] Wayne Wolf, "Modern VLSI Design-A systems approach", Prentice Hall-Inc., pp. 228-230, 1994.
- [7] Carver Mead and Lynn Conway, "Introduction to VLSI system", Addison-wesley publishing company, pp 150-154, 1980.
- [8] Abdellatif Bellaouar and Mohamed I. Elmasry, "Low Power Digital VLSI Design", Kluwer academic publishers, pp 451-454, 1995.
- [9] Makoto Suzuki, et al., "A 1.5ns 32-b CMOS ALU in Double Pass-Transistor Logic", IEEE J. Solid-State Circuits, Vol. 28, No. 11, November. 1993.
- [10] Yong Surk Lee et al, "Fast and gate-count efficient arithmetic logic unit", IEE Electronics letters, Vol. 32, No. 23, November, 1996.
- [11] 공배선, "고성능 마이크로프로세서를 위한 32-Bit Data-Path의 설계", 석사학위 논문, 한국과학기술원, 1991
- [12] 유범선, 이중석, 이기영, 조태원 "Switched capacitance 감소를 통한 저전력 16비트 ALU 설계", 전자공학회 논문지 제37권 SD편, 제1호, 2000
- [13] Anand Raghunathan et al, "Register Transfer Level Power Optimization with Emphasis on Glitch Analysis and Reduction", IEEE transactions on CAD of integrated circuit and systems, Vol. 18, No. 8, August, 1999.
- [14] Hema Kapadia et al, "Reducing Switching Activity on Datapath Buses with Control-Signal Gating", IEEE J. of Solid-State Circuits, Vol. 34, No. 3, pp 405-414, March, 1999.
- [15] C. Lemonds and S. Shetti, "A Low Power 16 by 16 Multiplier Using Transition Reduction Circuitry", International Workshop on Low Power Design, pp. 139-142, 1994.
- [16] Edwin de Angel and Earl E. Swartzlander, "Techniques for Low Power Paralell Multipliers", IEEE Computer Society TCVLSI Technical Bulletin, Spring, 2000.
- [17] Wei-Chang Tsai et al, "Accurate Logic-Level Power Simulation Using Glitch Filtering and Estimation", Proceedings of the IEEE Asia Pacific Conference on Circuit and Systems '96, 1996.
- [18] Alain Guyot and Selim Abou-Samra, "Low Power CMOS Digital Design," Proceedings of the Tenth International Conference on Microelectronics, pp. IP 6-13, 1998.
- [19] Dirk Rabe and Wolfgang Nebel, "Short Circuit Power Consumption of Glitches", In proceedings of the International symposium on Low Power Electronics and Design, 1996.
- [20] T. Shikata et al., "A Single-Chip Low Power DSP/RISC CPU with 0.25 μ m CMOS Technology", IEEE Custom Integrated Circuit Conference, pp. 123-126, 1998.
- [21] Luca. Benini et al., "Glitch Power Minimization by Selective Gate Freezing", IEEE Transactions on VLSI Systems, Vol. 8, No. 3, June, 2000.
- [22] 경북대학교, "DSP Macro Library개발에 관한 연구," ASIC 기반기술과제 3차년도 보고서, pp. 47-48, 1996
- [23] DSP/MSP Product Reference Manual, Analog Devices, 1995.
- [24] T. P. Kelliher, R. M. Owens, M. J. Irwin, and T. T. Hwang, "ELM-A Fast Addition Algorithm Discovered By a Program" IEEE Transactions on Computers, Vol. 41, No. 9, September, 1992.

- [25] C. Nagendra, M. J. Irwin, and R. M. Owens, "Area-Time-Power Tradeoffs in Parallel Adders" IEEE Transactions on Circuits and Systems-II Analog and digital signal processing, Vol. 43, No. 10, October, 1996.
- [26] John L. Hennessy and David A Patterson, "Computer architecture A Quantitative Approach", Morgan Kaufmann Publishers, pp. D-15~ D-22, 1996.
- [27] Digital Signal Processing Applications with the TMS320 Family - Theory, Algorithm and Implementations - , Digital Signal Processor Products, Texas Instruments, 1990.
- [28] TMS320C2x User's Guide, Digital Signal Processor Products, Texas Instruments, 1993.

 저 자 소 개

柳 凡 善(正會員) 第 37卷 SD編 第 1號 參照
 현재 충북대학교 대학원 전자공학과 박사과정

李 奇 榮(正會員) 第 37卷 SD編 第 1號 參照
 현재 충북대학교 전기전자공학부 교수

李 成 鉉(正會員)
 대전산업대 전자공학과 졸업(공학사). 충북대학교 대학원 전자공학과 졸업(공학석사). 충북대학교 대학원 전자공학과 박사과정 재학중. 현재 조폐공사 연구소 개발 연구실 선임연구원. 관심분야 : 저전력 회로설계, DSP 구조

趙 泰 元(正會員) 第 37卷 SD編 第 1號 參照
 현재 충북대학교 전기전자공학부 교수