論文95-32A-12-12

동기식 기억소자를 위한 레지스터를 이용한 병렬 파이프라인 방식

(Register-Based Parallel Pipelined Scheme for Synchronous DRAM)

宋 昊 俊 *

(Ho Jun Song)

요 약

최근 시스템의 성능이 향상되고 동작 속도가 빨라짐에 따라 외부의 클럭 신호와 동기되어 연속적으로 데이터를 출력할 수 있는 동기식 기억소자(Synchronous DRAM)가 등장하고 있다. 그러나 기존의 동기식 기억소자의 경우 버스트 리드(Burst Read) 동작을 위하여 컬럼 패스가 여러 단 직렬 형태로 나누어져 있으므로 동작 주파수가 증가하고 컬럼 래이턴시(CAS Latency)가 커질 경우 새로운 단(Stage)이 추가되어야 하고 이것으로 인한 기억소자 면적의 증가와 내부 신호들간의 시간 제어가 어려워지는 단점이 있었다. 본 논문에서는 면적의 증가를 최소화하고 내부 신호들의 시간 제어를 용이하게 함으로써 높은 동작 주파수에서 동작할 수 있는 레지스터를 이용한 병렬 파이프라인 방식을 제안한다. CAS Latency가 N인 경우 (N-1)개의 레지스터가 데이터 버스 라인과 데이터 출력 버퍼사이에 위치하여 연속적으로 들어오는 데이터를 순차적으로 저장하고 N번째 클럭 신호부터 데이터 출력 버퍼를 제어하여 정해진 버스트 길이동안 연속적으로 데이터를 내보낸다. 본 방식에 따르면 컬럼 데이터 패스가 여러 단으로 나누어져 있지 않으며 데이터가 마지막 출력 버퍼단에서만 제어되므로 동기식 기억소자의 버스트 리드(Burst Read) 동작을 용이하게 구현할 수 있다. 0.32μm-technology 4-Bank 64M 동기식 기억소자(SDRAM)의 컴퓨터 시뮬레이션 결과 본 방식에 따른 동기식 기억소자의 경우 CAS Latency N = 4일 경우 N = 3에 비하여 면적 증가는 0.1% 이하이며 버스트 리드 동작은 200MHz까지 성공적으로 동작함을 보인다.

Abstract

Recently, along wtih the advance of high-performance system, synchronous DRAM's (SDRAM's) which provide consecutive data output synchronized with an external clock signal, have been reported. However, in the conventional SDRAM's which utilize a multi-stage serial pipelined scheme, the column path is divided into multi-stages depending on CAS latency N. Thus, as the operating speed and CAS latency increase, new stages must be added, thereby causing a large area penalty due to additinal latches and I/O lines. In the proposed register-based parallel pipelined scheme, (N-1) registers are located between the read data bus line pair and the data output buffer and the coming data are sequentially stored. Since the column data path is not divided and the read data is directly transmitted to the registers, the busrt read operation can easily be achieved at higher frequencies without a large area penalty and degradation of internal timing margin. Simulation results for 0.32um-Tech. 4-Bank 64M SDRAM show good operation at 200MHz and an area increment is less than 0.1% when CAS latency N is increased from 3 to 4.. This pipelined scheme is more advantageous as the operating frequency increases.

* 正會員, 忠南大學校 電子工學科

(Dept. of Electronics Eng., Chungnam National University)

接受日字:1994年4月20日,수정완료일:1995年11月22日

I. 서 론

최근 시스템의 성능이 날로 향상되고 동작 스피드가 빨라짐에 따라 다이나믹 기억소자 (DRAM)의 동작 속도도 그에 따라 고속화가 요구된다. 그러나 일반적으로 기억소자의 경우 데이터를 감지 및 증폭하여 데이터 출력 버퍼까지 도달하는 데 걸리는 시간이 시스템의 성능과 보조를 맞추어 단축되지 못하고 이는 곧 전체 시스템의 성능을 향상하는 데 있어서 제한적인 요소가 되어왔다. 이러한 문제점을 극복하고자 고속 기억소자 즉, 캐쉬 기억소자(Cache DRAM)[1], Rambus DRAM (RDRAM)[2], 동기식 기억소자 (Synchronous DRAM)[3-6] 등이 활발히 연구되어 왔다. 그 중에서도 동기식 기억소자(이하, SDRAM 이라 칭함)의 경우 외부 클럭 신호와 동기되어 높은 동작 스피드로 연속적인 데이터를 출력할 수 있을 뿐만 아니라 컴퓨터의 주 기억 장치 (Main Memory)로도 쓰일 수 있으므로 더욱 더 각광을 받고 있으며 곧 상용화되기 위한 전단계에 접어들었다. SDRAM의 가장 중요한 동작이라고 말할 수 있는 버스트 리드 (Burst Read) 동작은 외부 클럭 신호와 동기되어 연속적으로 데이터를 출력하는 동작으로서 여러 개의 뱅크(Bank) 중 최소한 하나의 워드 라인 (Word Line)이라도 활성화(Activated)되어 있을 때 컬럼 어드레스와 버스트 리드 명령이 들어오면 주어진 CAS latency와 Burst Length에 따라 예를 들면, CAS Latency = 3, Burst Length = 4 인 경우 상기 컬럼 어드레스 및 버스트 리드 명령이 들어온 클럭을 기준으로 3번째 클럭부터 4개의 데이터가 연속적으로 클럭 신호에 맞추어 출력된다. 즉 CAS latency = N, Burst length = M인 경우 컬럼 어드레스 및 버스트 리드 명령이 들어온 후 N번째 클럭부터 M개의 데이터가 연속적으로 출력된다. 그림 1은 기본적인 버스트 리드 동작을 설명하는 타이밍을 보여준다.

그림 1의 버스트 리드 동작을 위하여 기존의 SDRAM의 경우 컬럼 데이터 패스를 N개의 단(Stage)으로 나누어 파이프라인 형태로 구성한다. 즉 매 한 클럭마다 데이터가 한 단씩 이동하여 N번째 클럭부터 데이터가 순차적으로 출력되는 구조로서 기본적인 CAS latency = 3인 경우의 대략적인 블럭이 그림 2에 나타나있다 (경우에 따라서는 각 단의 구성이 조금씩 틀릴 수도 있음).

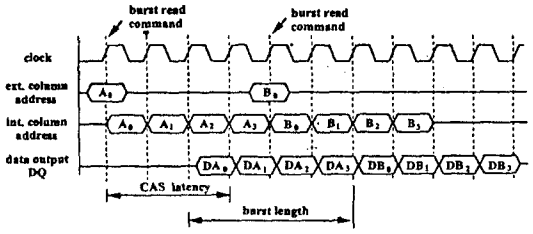


그림 1. 버스트 리드 동작의 설명도 (CAS Latency = 3, Burst Length = 4)
Fig. 1. Timing diagram of burst read operation for CAS latency = 3 and burst length = 4.

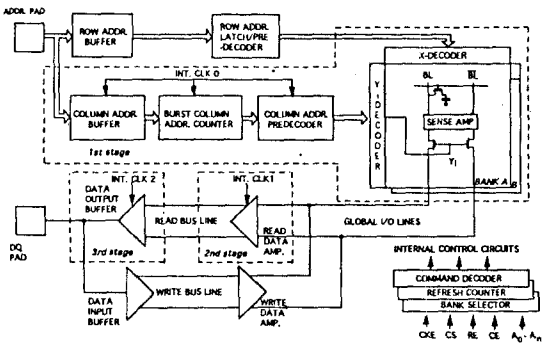


그림 2. 기존의 다단 파이프라인 방식을 이용한 SDRAM의 블럭도
Fig. 2. Block diagram of the conventional multi-stage pipelined SDRAM.

이러한 기존의 직렬 파이프라인 구조에서는 한 클럭 주기동안 각 단의 데이터가 다음 단으로 전달되어야 하므로 동작속도는 여러 단 중에서 가장 시간 지연이 많은 최장단 (Longest Stage)에 의해 제한된다. 따라서 이러한 구조에서는 각 단의 시간 지연이 동일하게 나누어져야 하므로 컬럼 패스상에서 불필요한 래치 (Latch)회로 및 I/O 라인들에 의하여 면적의 증가를 유발하고 내부 제어 신호들의 시간적인 여유도 (Timing Margin)가 떨어지는 문제점이 있었다. 더우기 동작속도가 더욱 증가하면 이때 컬럼 패스상의 데이터가 전달되는 시간은 각 기억소자마다 일정한 값을 가지므로 CAS latency는 더욱 증가하게 되고 이로인해 새로운 단이 추가되어야 하므로 각 단의 균등분할을 위한 컬럼 패스의 구성은 더욱 어려워지고 이로인한 부수적인 래치 및 I/O라인들의 추가에 따른 면적의

증가는 급격하게 증가하고 내부 신호들간의 시간제어는 더욱 더 어려워지게 된다.

이러한 문제점을 해결하고자 최근에 레지스터를 이용한 웨이브 파이프라인 방식[7-9] 및 FIFO (First In First Out) 방식[10]들이 발표되어 큰 주목을 끌고 있다. 이러한 방식을 활용하면 컬럼 패스를 여러 단으로 나누지 않을 수 있고 동작속도가 증가함에 따라 간단히 레지스터를 추가함으로서 CAS latency를 증가시킬 수 있으므로 큰 면적의 증가없이 고속의 SDRAM을 구현할 수 있다.

본 논문에서는 (N-1)개의 레지스터를 이용하고 입/출력 제어가 용이한 파이프라인 방식을 제안하고 제안된 방식을 이용하여 SDRAM의 버스트 리드 동작을 기술하였다. 본 방식에 따른 SDRAM을 컴퓨터 시뮬레이션을 통해 현재 설계중인 0.32μm-Tech. 4-Bank 64M SDRAM에 적용하여 200MHz까지 버스트 리드 동작이 수행됨을 확인하였다.

II. 레지스터를 이용한 병렬 파이프라인 방식

그림 3은 본 방식에 따른 SDRAM의 개략적인 블럭도를 보여준다.

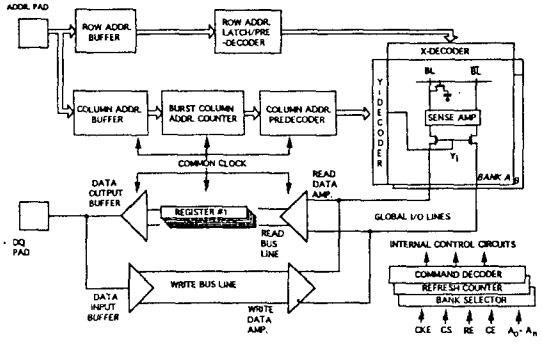


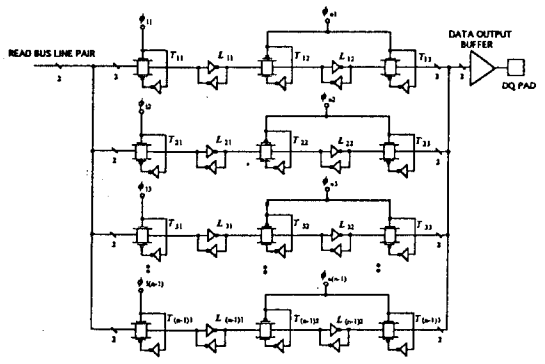
그림 3. 제안된 방식의 파이프라인을 이용한 SDRAM의 블럭도

Fig. 3. Block diagram of a proposed register-based parallel pipelined SDRAM.

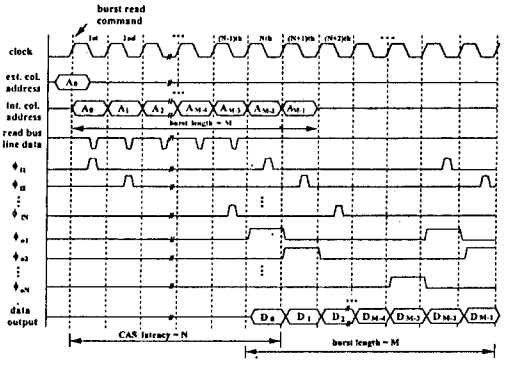
본 방식에서는 컬럼 데이터 패스가 N개의 단으로 나누어지는 대신 (N-1)개의 레지스터가 리드 버스 라인 쌍과 데이터 출력 버퍼사이에 병렬로 연결되어 있다. 처음에 버스트 리드 명령과 시작 어드레스에 해당하는 첫번째 컬럼 어드레스가 클럭의 상승에지에서 입력된 후 첫번째 컬럼 어드레스에 해당하는 데이터가 리드 버스 라인 쌍에 도착하면 리드 버스 라인 쌍과 데이터 출력 버퍼 사이에 병렬로 연결되어 있는 레지스터 중 첫번째 레지스터에 저장된다. 이어서 두번째 컬럼 어드레스(첫번째 컬럼 어드레스를 내부 어드레스 카운터에 받아들인 후 두번째 컬럼 어드레스부터는 내부 컬럼 어드레스 카운터를 하나씩 증가시키면서 발생됨)에 해당하는 데이터가 도착하면 두번째 레지스터에 직접 저장된다. 이러한 방식으로 (N-1)개의 데이터가 모두 레지스터에 저장된 다음, N번째 클럭 사이클에서 처음에 저장되었던 첫번째 레지스터의 데이터가 출력 버퍼를 통해 출력되고 그와 동시에 새로 들어오는 N번째 데이터가 첫번째 레지스터에 다시 저장된다. 다음 (N+1)번째 클럭 사이클에서는 두번째 레지스터에 저장되었던 데이터가 데이터 출력 버퍼를 통해 출력되고 동시에 새로 들어오는 데이터가 다시 두번째 레지스터에 저장된다. 이러한 동작을 주어진 Burst Length M 사이클동안 반복하여 출력단에서는 버스트 리드 명령이 입력되고 N번째 클럭부터 M개의 사이클동안 연속적인 데이터를 출력단에서 얻을 수 있다. 본 방식의 파이프라인 SDRAM의 버스트 리드 동작은 컬럼 데이터 패스가 N단으로 나누어져 있지 않고 데이터가 데이터 출력 버퍼 앞 단의 레지스터에 바로 순차적으로 저장되므로 고속동작에서 기존의 N단 직렬 파이프라인 처럼 가장 긴 단에 의하여 제약을 받지 않는다. 이는 데이터가 직접 데이터 출력 버퍼 앞 단까지 전달되기 때문에 사실상 여러 단이 동일한 시간지연을 갖는 균등분할한 것과 같은 효과를 얻을 수 있기 때문이다. 이러한 방식의 또 하나의 장점은 기존 방식이 CAS Latency N이 증가하면 컬럼 패스상에 새로운 단을 추가하여야 하므로 내부 각 단의 구성이 더욱 복잡해지고 내부 제어 신호 및 래치회로들이 추가되는 반면 본 방식에서는 컬럼 패스 구성의 변화없이 레지스터만 추가하여 CAS latency의 증가에 대응할 수 있다.

제안된 방식에 따른 동작을 로직도 및 파형도를 통하여 살펴보면 그림 4(a)-(c)에 나타난 바와 같다.

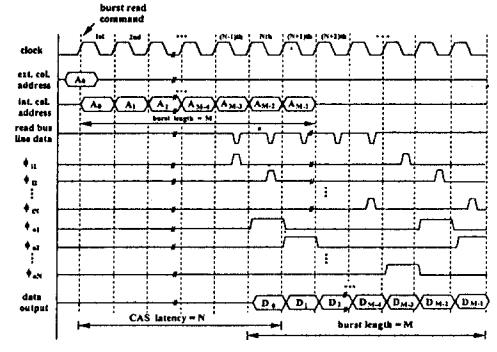
그림 4(a)와 4(b)에서, 최소한 한개의 워드 라인이 활성화되어 있다고 가정하고 동작 주파수가 낮은 영역이라고 가정하고 버스트 리드 명령과 첫번째 컬럼 어드레스가 첫번째 클럭 사이클에서 입력되면 이 번지에 해당하는 데이터가 리드 버스 라인 쌍에 도달하고 이 데이터는 첫번째 레지스터 SR_1에 저장된다.



(a)



(b)



(c)

그림 4. 레지스터를 이용한 파이프라인 방식
(a) 로직도
(b) 저주파수에서의 타이밍도
(c) 고주파수에서의 타이밍도

Fig. 4. Register-Based Parallel Pipelined Scheme.
(a) Logic Diagram
(b) Timing Diagram at low frequencies
(c) Timing Diagram at high frequencies

즉 이때 입력 제어 신호 ϕ_{i1}은 'H'로써 첫번째 전달 게이트 T_{11}이 턴-온 (turn-on)되고 출력 제어 신호 ϕ_{o1}은 'L'이므로 역시 두번째 전달 게이트 T_{12} 도 턴-온 상태이므로 첫번째 데이터는 첫번째 레지스터의 두번째 래치 L_{12}까지 전달된다. 이때 입력 제어 신호 ϕ_{i1}은 기존의 기억장치에서 쓰이는 통상의 방법처럼 인버터 체인등을 이용한 시간 지연 회로를 사용하여 데이터가 리드 버스 라인 쌍에 도달하는 시간을 맞추어 잠시동안 펄스 형태로 만들어지고 출력 제어 신호 ϕ_{o1}은 첫번째 레지스터의 데이터가 데이터 출력 버퍼에 전달될때만 'H'가 된다. 이어서 내부 컬럼 어드레스 카운터에서 발생된 두번째 컬럼 어드레스에 해당하는 데이터가 리드 버스 라인 쌍에 도착하면 앞의 방법과 마찬가지로 두번째 레지스터 SR_2의 두번째 래치 L_{22}까지 전달되어 저장된다. 즉 입력 제어 신호 ϕ_i는 매 데이터가 리드버스 쌍에 도착할때마다 펄스 형태로 도착하는 시간에 맞추어서 순차적으로 턴-온되고 출력 제어 신호 ϕ_o는 해당하는 레지스터의 데이터가 출력될때만 턴-온된다. 이러한 방식으로 (N-1)번째 클럭 사이클까지 데이터가 레지스터들 SR_1-$SR_{(N-1)}$의 두번째 래치들 L_{12}-$L_{(N-1)2}$까지 순차적으로 저장된 후 N번째 클럭 사이클에서 첫번째 레지스터 SR1의 두번째 래치의 데이터 D0를 출력하기 위하여 첫번째 출력 제어 신호 ϕ_{o1}이 'H'가 된다. 그러면 첫번째 레지스터의 세번째 전달 게이트 T_{13}가 턴-온되고 첫번째 데이터 D_0가 데이터 출력 버퍼를 통하여 데이터 패드 (Data pad DQ)에 나타난다. 이동안, 출력 제어 신호 ϕ_{o1}이 'H'이므로 첫번째 레지스터의 두번째 전달 게이트 T_{12}는 턴-오프가 되어 첫번째 래치L_{11}과 L_{12}가 분리되어 새로 들어오는 N번째 데이터 DN-1을 첫번째 래치회로 L_{11}에 저장한다. 마찬가지로 (N+1)번째 클럭 사이클에서는 두번째 출력 제어 신호 ϕ_{o2}가 'H'가 되어 두번째 레지스터의 세번째 전달 게이트 T_{23}가 턴-온되어 두번째 레지스터 SR_2의 두번째 래치 L_{22}에 있던 두번째 데이터 D_1이 데이터 출력 버퍼를 통하여 출력되고 마찬가지로 두번째 전달게이트 T_{22}가 턴-오프 되어있어 (N+1)번째 데이터 DN이 두번째 레지스터 SR_2의 첫번째 래치 L_{21}에 전달되어 저장된다. 한편, N번째 클럭 사이클이 지나면 첫번째 출력 제어 신호 ϕ_{o1}이 즉시 'L'가 되어 첫번째 레지스터 SR_1의 첫번째 래치의 데이터 D_{N-1}이 두번째 전달게이트 T_{12}를 통하여 두번째 래치 L_{12}로 바로 전달되어 다음 N번째 클럭 사이클에서 즉

시 데이터를 내보내고 다시 데이터를 받아들일 준비를 한다. 이와같은 동작이 Burst Length M 사이클동안 반복되어 출력단에 N번째 클럭 사이클부터 연속적으로 M개의 데이터가 출력되게 된다. 요약하면 리드 버스 쌍에 도달하는 데이터를 순차적으로 레지스터에 저장하고 저장된 데이터를 순차적으로 출력시키면서 새로운 데이터를 받아들이는 동작을 반복하는 것으로써 데이터의 입출력을 출력 버퍼 앞 단에서만 순차적으로 제어하는 것이다.

그림 4(b)의 파형도는 동작 주파수가 낮아 한 클럭 주기안에 데이터가 리드 버스 쌍에 도달하는 경우를 나타나는 것이다. 그러나 동작 주파수가 증가하여 데이터가 몇개의 클럭이 지나고 들어오는 경우에는 그림 4(c)에 보인 바와 같이 실제로 N번째 클럭 이전에 (N-1)개의 레지스터가 모두 데이터로 채워지지 않을 수도 있다. 그러나 입력은 들어오는 대로 레지스터들에 순차적으로 순환하면서 채워지고 출력단은 정해진 N번째 클럭부터 순차적으로 순환하면서 출력되므로 동작은 앞서 기술한 방법과 유사하다. 첫번째 데이터가 N번째 클럭 사이클 이전 즉, (N-1)클럭 사이클의 어느 시점에선가 들어온다고 가정하면 (사실상 이 조건 즉, 컬럼 어드레스 디코더부터 리드 버스 쌍까지의 데이터 전달 시간 $_{td}$ 가 (N-1) x 1/f $_{clock}$ 가 보다 작아야만 N번째 클럭에서 첫데이터가 출력될 수 있으므로 td 〈 (N-1) x 1/f $_{clock}$ 조건에서 최대 동작 주파수가 정해지고 동작 주파수가 증가하면 td는 일정하므로 CAS Latency가 증가하게 된다) 첫번째 데이터가 (N-1)번째 클럭에서 첫번째 레지스터에 저장되고 바로 다음 N번째 클럭에서 다시 출력되고 이 N번째 사이클동안 두번째 데이터는 두번째 레지스터에 저장되게 된다. 다음 (N+1)번째 사이클에서 두번째 레지스터의 데이터가 출력되면서 동시에 이 사이클 동안 세번째 데이터가 세번째 레지스터에 들어가고 다음 사이클에서 출력된다. 다시 말해서 몇번째 클럭에서 첫번째 데이터가 들어오는 가에 따라서 N번째 클럭이전에 몇개의 레지스터에 데이터가 저장되는 지가 결정되나 결국 입력 제어신호가 순차적으로 순환하고 출력 제어 신호가 순환하므로 동작은 유사하게 된다. 그러나 동작 주파수에 관계없이 저주파수나 고주파수나 항상 동작하여야 하므로 최소한 (N-1)개의 레지스터가 필요하다. 이는 기존의 다단 직렬 파이프라인 구조에서도, 만약 동작 주파수가 하나로 결정난다면, 예를 들어 (N-1)번째 클럭 사이클안에 데이터가 리드버스 라인에 전달된다면 사실상 기존의 방식에서도 N개의 단으로 나누지 않고 두 단으로 나누어 출력 버퍼에서만 클럭 제어를 받아 출력시키면 되는 것과 동일한 원리이다. 그러나 사실상 모든 주파수에서 동작을 하여야 하므로 기존의 방식이 N개의 단으로 나누어져야 되는 것으로 생각 할 수 있다.

Ⅲ. 컴퓨터 모의 실험 및 결과

그림 5는 하나의 레지스터의 실제적인 구현 회로를 나타낸다.

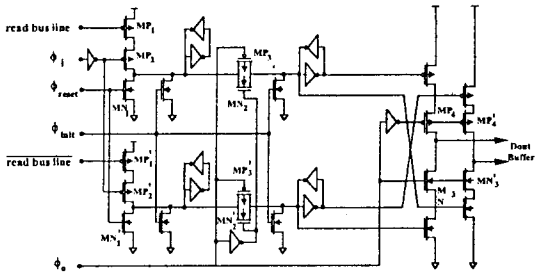


그림 5. 하나의 레지스터의 실제 구현 회로
Fig. 5. Real implementation of one register.

즉, 이러한 레지스터가 CAS Latency에 따라 (N-1)개 병렬로 연결된다. 첫번째 전달게이트는 MP_2와 MP_2'로 구성되어 있으며 리드 버스 쌍은 MP_1과 MP1′에 연결되어 있다. 데이터가 전달되었을 때 MP_2와 MP_2'가 입력 제어 신호 ϕi에 의하여 턴-온 된다. 이때 ϕ_i 신호는 리드 데이터에 데이터가 전달되는 시간에 동기되어 펄스 형태로 구동된다. 두번째 전달게이트는 MP_3, MN_2와 MP_3', MN_2'로 구성되고 세번째 전달게이트는 MP_4, MN_3 와 MP_4', MN_3'로 구성된다. 첫번째 전달게이트가 일반적인 두번째 전달게이트와는 조금 다른 구성을 가지고 있다. 그 이유는 다음과 같다. 일반적으로 SDRAM에서 리드 버스 쌍에 도달하는 데이터는 어느정도의 시간 폭을 가진 펄스 형태로 되고 리드 버스 쌍은 전원 전압 레벨로 프리-챠아지(Precharge)되어 다음 데이터를 다시 빠른 속도로 Evaluation할 준비를 한다. 그러므로 첫번째 전달게이트의 구성이 두번째의 일반적인 구성을 가지게 되면 펄스 형태의 데이터가 도착했을 때 입력 제어 신호는 정확하게 동기되어 데이터 펄스 폭 보다 작은 폭을 가

지고 전달게이트를 턴-온 되어야 한다. 만약 그렇지 않으면, 즉 입력 제어 신호의 펄스 폭이 더 크거나 살짝 지연이 더 되면 데이터가 다시 프리-챠아지되어 데이터를 잃어버리므로 첫번째 래치에 데이터를 저장할 수 없다. 따라서 입력 제어 신호의 시간적 제어가 어려워진다. 그러나 그림 5에 나타난 첫번째 전달게이트 구성은 그러한 어려움이 없이 어느 정도 입력제어 신호가 데이터 신호보다 큰 폭을 가지고만 턴-온되면 공정 변화나 지연시간이 조금 변해도 데이터를 유실하지 않고 래치에 전달할 수 있기 때문이다. 이 전달게이트는 한 쪽 방향으로만 데이터가 Evaluation되므로 다음에 데이터가 오면 리셋 신호 φreset 에 의하여 리셋시키킨 후 다시 래치에 새로운 데이터를 저장한다. φinit은 전원을 켠 후 초기조건을 잡아준다. 그림 6은 본 방식에 의한 파이프라인 0.32um-Tech. 4-Bank 64M SDRAM을 CAS Latency = 4, Burst Length = 7, Frequency = 200MHz, Temp. = 300 K, Vcc = 3.3 V 조건에서 Data '0-1-0-0-1-0-1'를 미리 셀에 저장해놓고 컴퓨터 시뮬레이션을 수행한 결과이다. 시뮬레이션은 0.32μm 공정(3 Polysilicon/ 1 Policide/ 2 metal)에서 추출된 회로변수를 BSIM 모델로 HSPICE 에서 수행하였다.

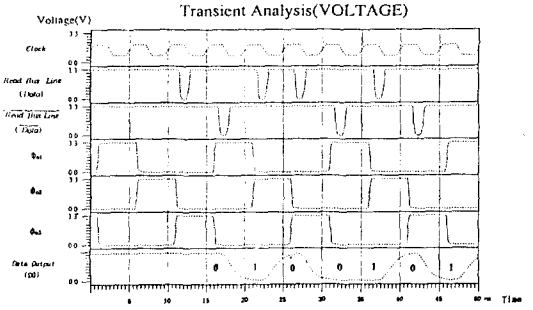


그림 6. 본 방식에 따른 SDRAM의 모의실험 결과
Fig. 6. Simulation results of the proposed pipelined SDRAM.

클럭 신호는 5ns 마다 0.8V 의 크기를 가지고 토글링(Toggling)한다. 버스트 리드 명령이 0ns에서 입력된 후 컬럼 디코더에서 하나의 컬럼을 선택하고 Global I/O 라인과 리드 데이터 증폭기를 거친후 11ns 후에 리드 버스 라인 쌍에 데이터가 전달된다. 이때 데이터는 약 1.6ns의 폭을 갖는다. 두번째 데이터는 두번째 클럭에서 새로운 컬럼 어드레스(두번째부터는 내부 컬럼 어드레스 발생기에서 만들어진다.)에 해당하는 컬럼이 선택되고 계속해서 데이터가 전달되어 오고있는 중이므로 매 클럭 주기마다 즉, 5ns마다 데이터가 계속 데이터가 리드 버스 라인 쌍에 나타난다. 이러한 데이터는 순차적으로 순환하면서 레지스터에 저장된다.(입력 제어 신호는 순차적으로 거의 동일한 펄스 폭을 동일한 타이밍에 인가된다. 그림 6에는 도시하지 않음). CAS latency가 4이므로 4번째 클럭 사이클부터 데이터는 순차적으로 출력된다. 외부 클럭이 'H'로 올라간 후 출력 제어 신호들 φo1, φo2, φo3은 각각 신호를 순차적으로 인가하기위한 순환 로직회로를 거치므로 외부 클럭 신호가 인가되고난 후 1.1ns 후에 각 레지스터의 두번째 및 세번째 전달게이트에 인가되고 각 데이터는 그로부터 약 2.7ns후에 데이터 출력 버퍼를 거쳐 데이터 패드에 나타난다. 즉, 리드 데이터는 버스트 리드 명령이 들어온 후 3클럭 + 3.8ns후에 첫번째 데이터가 출력된 후 매 클럭 (5ns) 마다 데이터가 출력된다.

한편 외부에서 주어진 컬럼 어드레스와 내부에서 발생된 컬럼 어드레스가 내부 로직회로에서는 동일하게 취급되고 각 레지스터의 입/출력은 순환하면서 동작하므로 새로운 컬럼 어드레스를 가진 버스트 리드 명령이 이전의 버스트 리드 동작중에 들어와도 버스트 동작은 중단되지 않고 새로운 어드레스부터 시작되는 버스트 동작이 연속해서 출력된다.

Ⅳ. 결 론

동기식 기억소자에서 (N-1)개의 레지스터를 이용한 파이프라인 방식이 제안되었고 컴퓨터 모의 실험을 통해 200MHz에서 동작이 됨을 확인하였다. 본 방식에 따른 파이프라인 Scheme을 이용할 경우 동작 주파수가 증가하고 CAS Latency가 증가해도 레지스터만 더 병렬로 연결하고 단순히 입/출력 제어 신호를 순환시켜 줌으로써 버스트 리드 명령을 구현할 수 있으므로 큰 면적의 증가 없이 SDRAM을 만들 수 있고 동시에 내부 제어 신호의 Timing 제어가 용이하다. 기존의 다단식 직렬 파이프라인 방식이 CAS Latency가 증가하면 부가적인 래치 및 데이터 버스 신호의 추가로 인해 면적이 크게 증가함에 반해 (CAS Latency가 4일 경우 CAS Latency 3일때 비해 16M/64M SDRAM의 경우 약 3-5% 증가) 본 방식에서의 면적 증가는

약 0.1% 이하이다. 한편 상기와 같은 고속 SDRAM의 경우 출력단에 DLL(Delay Locked Loop)를 이용하여 외부 클럭 신호에서 부터 데이터 출력 버퍼까지의 시간 지연을 보상하여 외부 클럭 신호가 인가된 후 바로 데이타가 출력 될 수 있는 방법이 최근 많이 연구되고 실용화 되고있으며 그에 대한 연구도 필수적이라 할 수 있다.

감사의 글

※ 본 논문의 수행에 있어서 컴퓨터 시뮬레이션 및 제반사항을 지원하여 주신 김정필, 오종훈, 이재진씨, 안승환 부장님 및 황인석 전무님이하 모든 설계실원 여러분께 감사드립니다.

참 고 문 헌

[1] K. Dosaka et al., "A 100-MHz 4-Mb Cache DRAM with Fast Copy-Back Scheme," in 1992 IEEE ISSCC Dig. Tech. Papers, pp. 148-149, Feb. 1992.

[2] N. Kashiyama et al., "500Mbyte/sec Data-Rate 512kbits x 9 DRAM Using a Novel I/O Interface," in Proc. 1992 Symp. VLSI Circuits, pp. 66-67.

[3] Y. Takai, et al., "250Mbyte/s Synchronous DRAM Using a 3-Stage-Pipelined Architecture," IEEE J. Solid-State Circuits, vol. 29, no. 4, pp. 426-430, Apr. 1994.

[4] Y. Choi, et al., "16Mbit Synchronous DRAM with 125Mbyte/s Data Rate, "IEEE J. Solid-State Circuits, vol. 29, no. 4, pp. 529-534, Apr. 1994.

[5] A. Fujiwara, et al., "A 200MHz 16Mbit Synchronous DRAM with Block Access Mode," 1994 Symp. on VLSI Circuits, pp. 79-80.

[6] Y. Komada, et al., "A 150MHz 4-Bank 64M-bit SDRAM with Address Incrementing Pipeline Scheme," 1994 Symp. on VLSI Circuits, pp. 81-82.

[7] D. Wang, et al., "A Bipolar Population Counter Using Wave Pipelining ti Achieve 2.5x Normal Clock Frequency," ISSCC Digest of Technical Papers, pp. 56-57, Feb., 1992.

[8] W. Burleson, et al., "Wave-Pipelining: Is it Practical?," IEEE Proc, ISCAS, pp. 163-166, June, 1994.

[9] H. J. Yoo, et al., "A 150MHz 8-Banks 256M Synchronous DRAM with Wave Pipelining Methods," ISSCC Dig. Tech. Papers, pp. 250-251, Feb., 1995.

[10] M. Horiguchi, et al., "An Experimental 220MHz 1Gb DRAM," ISSCC Dig. Tech. Papers, pp. 252-253, Feb., 1995.

[11] K. Endo, T. Matsumura, and J. Yamada, "Pipelined, Time-Sharing Access Technique for an Integrated Multiport Memory," IEEE J. Solid-State Circuits, vol. 26, no. 4, pp. 549-554, Apr., 1991.

[12] B. Gunning et al, "A CMOS Low-Voltage-Swing Transmission-Line Transceiver," 1992 IEEE ISSCC Dig. Tech. Papers, Feb., 1992, pp. 58-59.

저 자 소 개

宋 昊 俊(正會員)

1962년 3월 26일생. 1985년 2월 서울대학교 공과대학 제어계측공학과 졸업 (공학사). 1986년 3월 ~ 1988년 2월 한국과학기술원 전기 및 전자공학과 졸업 (공학석사). 1988년 3월 ~ 1992년 8월 한국과학기술원 전기 및 전자공학과 졸업 (공학박사). 1992년 8월 ~ 1995년 2월 현대전자 반도체 제1연구소 근무. 1992년 3월 ~ 현재 충남대학교 전자공학과 전임강사. 주관심분야는 회로설계 및 소자모델링