

SPICE를 이용한 16-BIT ALU의 회로 해석 및 설계에 관한 연구

正會員 姜 熙 照*

A Study on the Analysis and Design of 16-BIT ALU by Using SPICE

Heau Jo KANG* *Regular Member*

要 約 빠른 설계 시간 및 재 설계 가능성 부여 등에 주안점을 두어 고성능의 단일 칩 16-bit data path를 설계하였다. 원칙적인 설계 방법의 체계적인 연구를 위하여 module 화의 개념을 근간으로한 설계방법을 도입하였으며, 이에 따라 각 내부 블록이 bus에 연결되어 독립적으로 동작하는 subsystem이 되도록 하였고 이를 결합하여 전체 시스템의 설계를 완성하였다. 시스템은 data path이다.

data path는 16-bit의 데이터를 처리하는 부분으로 ALU(Arithmetic Logic Unit), register file, barrel shifter 및 bus 회로로 구성된다. 이 회로에서의 게이트의 폭과 길이는 spice2를 사용하여서 결정하였다. 회로 시뮬레이션의 결과는 기대하였던 회로 특성과 잘 일치하였다.

ABSTRACT This paper present a new design concept of a single chip 16-bit data path using the concept of modular design, the whole system is divided into several blocks which can be operated as an independent system itself.

Making the internal blocks can act as a subsystem, it is possible to shorten design turn-around time, to be redesigned effectively, and to optimize the system performance. The designed system is data path. The data path is to manipulate 16-bit integer data. It is composed of arithmetic logic unit, register file, barrel shifter and bus circuit. The widths and lengths of gate in the circuit were determined using SPICE2. The results of circuit simulation were in good agreement with expected circuit characteristics.

I. 서 론

현대 반도체 공정 기술의 발달은 수십만개에서

수백만개의 트랜지스터 및 수동소자를 갖는 시스템을 단일칩으로 집적시킬 수 있게 하였다.^(1,2) 다시 말하여 메가 비트 수준의 메모리, 고성능 마이크로 프로세서, 실 시간 DSP(Digital signal processing) 칩등의 개발이 놀라운 속도로 진행되고 있으며, 이러한 칩등의 개발에 있어서 VLSI(Very Largy Scale Intergration) 설계 기술의

*東新工科大学 電子工學科
Dong-shin Engineering University
論文番號 : 90-21(接受1989. 11. 29)

정립과 개발을 필수로 한다. 요즘의 VLSI 설계 기술은 거의 대부분이 CAD(Computer Aided Design) tool에 의해 좌우된다고 하나 설계자의 경험이 전체 시스템의 성능, 설계, 시간등을 좌우한다. 이와 같은 설계 능력을 단시일내에 습득하는 것은 거의 불가능하며, 많은 연구와 경험이 밑바탕 되어야 한다. 본 논문에서는 16-bit 단일 칩 데이터 경로의 설계와 회로를 설계 해석하는데 있어서 필요한 기본회로의 분석 고찰을 하였고, 블록은 그림 1-1에 나타낸 것과 같이 arithmetic logic unit(ALU), Shifter, register array, two ports (Left port, Right port), BUS 회로, 입출력 회로로 구성하였다.

캐리 발생 회로는 Manchester-type carry chain 회로로 구성하여 지연 시간 및 면적을 줄였으며 회로 해석은 SPICE2(Simulation Program With Intergrated Circuit Emphasis) 프로그램을 이용하였다.

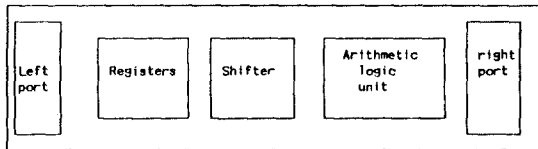


그림 1-1 일반적인 Data chip 블록도
Fig 1-1 General floor plan of the data path chip.

II-1. 기본회로 소자의 특성 분석 및 설계 원칙

전체 시스템을 구성하기 위해서는 공통적으로 쓰이는 기본회로 소자의 특성을 분석하고, 이에 따라서 설계 원칙을 다음과 같이 설정하였다. 게이트의 크기 및 트랜지스터의 모델설정은 설계사양 및 2 μ m p-Well CMOS공정에서 측출한 파라미터와 설계 원칙을 토대로 트랜지스터 모델을 설정하고⁽³⁾, SPICE 시뮬레이션 결과로 게이트 크기를 정하여 기본회로들을 설계하였다⁽⁷⁾. Inverter 및 transmission gate의 트랜지스터 크기를 다음과 같이 정하고 스위칭 특성을 조합하여 회로를 설계하였다.

i) 기본 inverter의 경우

$$\text{NMOS } L_n=2 \mu\text{m}, W_n=4 \mu\text{m}$$

$$\text{pmos } L_n=2 \mu\text{m}, W_p=4 \mu\text{m}$$

ii) Transmission gate 경우

$$\text{NMOS } L_n=2 \mu\text{m}, W_n=4 \mu\text{m}$$

iii) Depletion Mode gate 경우

$$\text{NMOS } L_n=12 \mu\text{m}, W_n=16 \mu\text{m}$$

II-2. Delay 추정방법

일반적으로 회로의 지연 계산은 시뮬레이션에 의해서 이루어지나, 시뮬레이션에 선행하여 대략적인 지연에 대한 추정이 필요하다. 이런 추정의 보편적인 방법으로 입력 캐패시턴스와 NMOS 트랜지스터의 유효 저항만을 이용한 RC딜레이 모델이 있으나 이 방법은 출력 소모 캐패시터의 영향을 무시하여 실제와는 큰 차이가 있다^(3,4) 본 연구에서는 출력소모 캐패시터를 추가한 지연의 추정 방법을 사용하였다. 즉 그림 2-1과 같은 회로에서의 지연은 R_{eff}, C_{in} 이 아니라 $R_{eff}, (C_{in}+C_{out})$ 이 된다는 것이다.

여기에서

$$R_{eff}=4 \frac{L}{W \mu C_{ox} B_{dd}}$$

$$C_{in}=C_{ox}$$

$$C_{out}=(1-2) C_{in} \text{이다.}$$

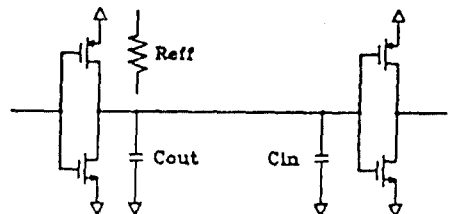


그림 2-1 delay 추정방법
Fig 2-1 delay deduction method

II-3. 최적화한 buffer

본 Inverter가 여러개의 게이트를 구동할 때

delay는 대략 fan-out의 수에 의해서 증가하게 된다. 이러한 경우에 중간에 보통 staged buffer를 얻도록 한다. buffer를 삽입한 total delay는 기존의 방법에 따르면 다음과 같다.

$$T = NfI = \ln(n)f / \ln(f)I$$

여기에서

$$N = \text{buffer stage} + 1$$

f = 각각의 buffer stage 사이의 게이트 사이즈비

$$n = C_L / C_{in} \text{을 나타낸다.}$$

다른 total delay의 변화는 그림 2-2와 같다.

f 가 e일 때 total delay는 최소가 된다. Fanout에 따른 stage의 갯수는

$$S = \text{Stage의 갯수} \ln(n) / \ln(f) - 1 \text{이 된다.}$$

설계시에 buffer stage 간의 gate area의 ratio를 e 로 할 경우 e 보다 큰 다른 ratio에 비해서 buffer stage가 증가함과 동시에 면적이 많이 차지하게 된다. 그런데, 실제의 Simulation 결과에서는 이러한 기존의 방법과 잘 맞지 않는다. 즉, ration가 e 일 때 보다는 e 보다 큰 4-5에서 근소한 차이 (n 이 e^7 정도일 때 수 nsec 정도)지만 minimum delay를 갖는 것으로 나타난다. 이러한 이유로서는 drain junction capacitance와 Cgd의 output capacitance에 미치는 영향을 무시했기 때문이다.

이의 분석은 다음과 같다. 그림 2-3은 buffer stage의 X 번째를 나타낸다.

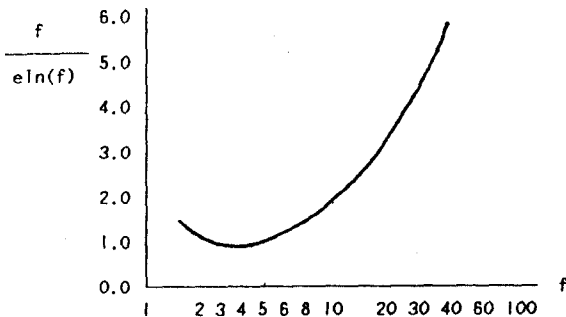


그림 2-2 크기 팩터 f 에 따른 시간 Penalty
Fig 2-2 Relation time penalty $f / e \ln(f)$ Versus size factor f

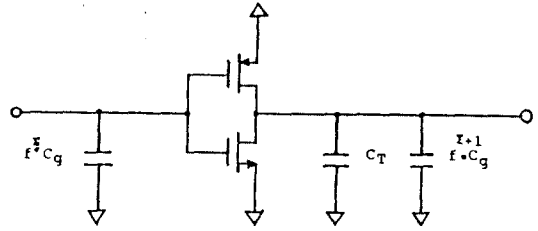


그림 2-3 buffer의 x번째의 stage

그림 2-3의 회로에서 C_t 는 drain의 junction Capacitance와 overlap capacitance Cdg의 Miller effect를 고려한 Capacitance이다.

$$C_t = K_{eq}(C_{dbn} + C_{dbp} + C_{jswn} + C_{jswp}) + 2(C_{dgn} + C_{dgp})$$

non-linear capacitance C_{ab} 를 linear Capacitance로 근간시키기 위한 것으로 K_{eq} 는,

$$K_{eq} = \frac{1}{V_2 - V_1} \frac{\phi_0}{1 - m} [(\phi_0 - V_2)^m - (\phi_0 - V_1)^m]$$

으로 된다.

ϕ_0 : built-in junction potential

$$\phi_0 = V_t \cdot \ln[N_e \cdot N_d / (n_i^2)]$$

V_2 : output low junction에 인가된 전압, -vol

V_1 : output high junction에 인가된 전압, -voh

m : grading coefficient

$$m = \begin{cases} 1/2 (\text{abrupt junction}) \\ 1/3 (\text{graded junction}) \end{cases}$$

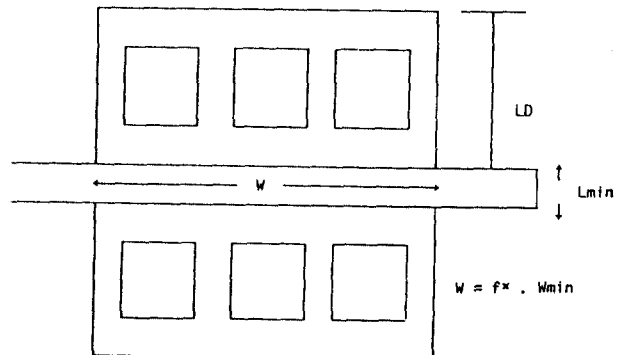


그림 2-4 Buffer의 nMOS부분 layout

C_t 의 값을 알기 위해서는 실제의 layout에 관한 정보가 필요한데 보통의 CMOS 공정은 그림 2-4와 같다.

위의 layout 정보를 이용하면

$$C_{tn} = K_{eq}(C'_{dbn} W_x L_d + C'_{jsw} L_d) + 2C'_{dgn} W_x$$

$$= a_n f^x C_{gn} + b_n f^x C_{gn}$$

가 된다.

여기에서

$$a_n = K_{eq}(C'_{dbn} / C'_{ox} + C'_{jsw} / C'_{ox} W_{min})(L_d / L_{min})$$

$$b_n = 2(C'_{dgn} / C'_{ox})(L_{min})$$

따라서 C_t 는 다음과 같다.

$$C_t = (a+b)f^x C_g$$

$$a = a_n = a_p$$

$$b = b_n = b_p$$

여기에서 a, b는 주어진 공정하에서는 상수이다.

total delay는

$$T = N(R_{min} / f^x)[(a+b)f^x \cdot C_g + f^{x+1} C_g]$$

$$= N(a+b+f)I$$

$$= I_n(n)(a+b+f) / I_n(f) \cdot I$$

minimum delay를 위한 f는

$$[(a+b+f) / I_n(f)]' = 0$$

$$I_n(f) = (a+b/f+1)$$

이때의 total delay는

$$T = I_n(n) \cdot f \cdot I \text{이다.}$$

II-4. Fanout 인수 결정

Fanout은 회로의 구동 능력을 결정하는 인수로써 회로설계의 중요한 파라미터의 하나이다. fanout을 측정하기 위해서 기본 inverter의 T_r 크기는 pMOS, nMOS, $W=8 \mu\text{m}$, $L=2 \mu\text{m}$ 로 정하여 최악조건하에서 fanout을 $W=4 \mu\text{m}$ $L=2 \mu\text{m}$ 로 결정하였다. 그림 2-5는 결정하기 위한 회로이다.

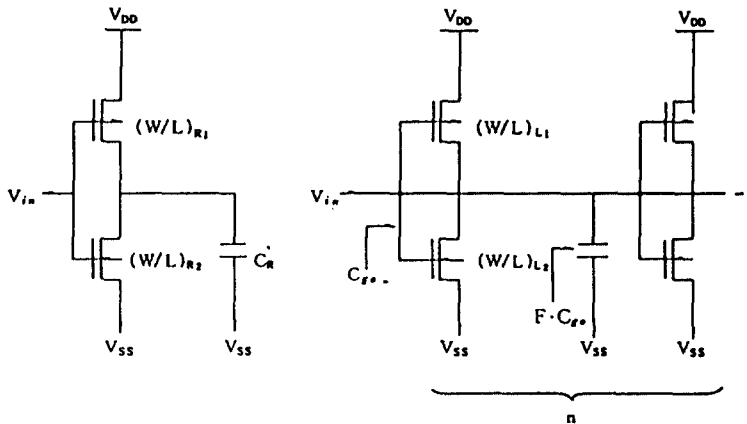


그림 2-5 Fanout을 결정하기 위한 회로.
Fig 2-5 Circuits for fanout determination.

그림 2-5에서 기본 inverter가 입력에 따라 PMOS, NMOS가 on, off 상태에 있다고 가정할 경우 fanout인수 F는 다음과 같이 계산된다.

Drain 전압 I_{ds} :

$$I_{ds} = \frac{\beta}{2} [(V_{gs} - V_t)(1 + \lambda V_{ds})] \quad (2.18)$$

여기서, λ 는 channel length modulation으로 $0.02 \text{ V}^{-1} \sim 0.04 \text{ V}^{-1}$ 값을 갖는다. 식(2.18)으로부터

드레인 전류는 β 에 비례하므로 다른 식이 성립한다.

$$I_g : I_L = \left(\frac{W}{L}\right) L = \frac{C_R}{t_r} : \frac{C_{G0} \cdot F}{t_m} \quad (2.19)$$

$$F = \frac{(W/L)_R \cdot t_m \cdot C_r}{(W/L)_R \cdot t_r \cdot C_{G0}} \quad (2.20)$$

여기서

$$C_{G0} = (W + \Delta W) (L + \Delta L) C_{ox} + C_L$$

t_m = measurement time

t_r = reference time

Ⅲ. 회로설계

Ⅲ-1. DATA Path chip의 설계

Data path는 16-bit data 처리를 하는 부분으로 크게 arithmetic logic unit(ALU) Register array, barrel shifter 및 두개의 BUS line 입출력화로 2개의 port로 나누어지며 그림 3-1과 같이 data path를 설계했다.^(4 ~ 6)

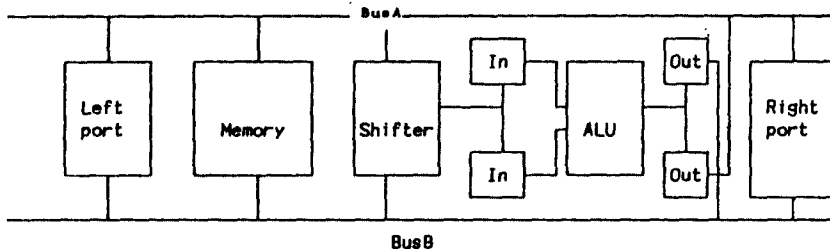


그림 3-1 Data path의 설계
Fig 3-1 General floor plan of the data path chip

Ⅲ-2. Arithmetic logic unit.

ALU(Arithmetic logic unit)는 Arithmetic operation 과 logic operation을 수행하는 부분으로

데이터 처리와 유효번지를 계산하는데 쓰인다. 이의 전체 구조는 그림 3-2와 같다.

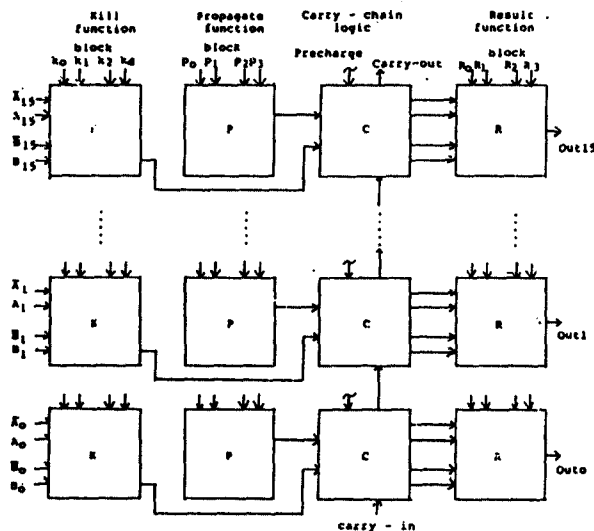


그림 3-2 16-bit ALU 구성도
Fig 3-2 Block diagram of a 16-bit ALU

그림 3-2에서와 같이 ALU는 P, K 및 R 블록, 입출력 및 캐리 발생 부분으로 크게 나누어진다. P, K 및 R 블록은 일종의 기능 블록으로 ALU에서 수행하는 명령에 따라 원하는 logic 값을 만들어 낸다.

Ⅲ-2-1, P, K 및 R block

P, K 및 R 블록의 내부 회로는 동일한 기본 기능 cell로 그림 3-3과 같다. 이 기능 cell은 두개

의 operand A, B에 대한 어떠한 동작도 가능한데 원하는 동작은 G0, G1, G2, G3의 제어 입력에 의해 결정된다. 따라서 이러한 P, K 및 R 블록을 사용하였을 때의 가장 큰 장점은 어떠한 명령이든지 P, K 및 R 블록에 인가되는 제어 입력만 결정 해주면 되므로 새로운 명령어의 추가가 아주 용이한 장점을 가지고 있다. P, K 및 R 블록의 G0, G1, G2, G3에 따른 동작의 종류는 표 3-1과 같다.

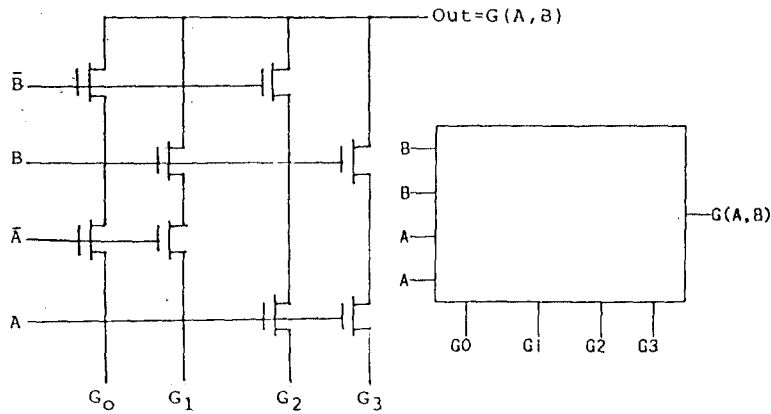


그림 3-3 P, K, R 내부 회로 블록도
Fig 3-3 General logic function block diagram for a transistor

표 3-1 P, K 및 R block.
Table 3-1 P, K and R block.

G3	G2	G1	G0	out	function
0	0	0	0		0
0	0	0	1		A, B
0	0	1	0		A, B
0	0	1	1		A
0	1	0	0		A, B
0	1	0	1		B
0	1	1	0		A, A+AB(A+B)
0	1	1	1		AB+AB+AB
1	0	0	0		AB
1	0	0	1		AB+AB(A B)
1	0	1	0		B
1	0	1	1		AB+AB+AB
1	1	0	0		A
1	1	0	1		AB+AB+AB
1	1	1	0		AB+AB+AB
1	1	1	1		1

Ⅲ-2-2. Carry 발생 블록

-ALU에서 산술 연산을 수행할 때 가장 많은 시간을 소모하는 부분이며 캐리를 발생시키는 방법에 따라 carry-look-ahead, manchester carry chain, carry-bypass, ripple carry, carry-select p-adder, brent-kung, QAC 등으로 크게 나누어진다.

대부분의 VLSI 시스템에서는 작은 면적 평이한 레이아웃(layout) 및 지연시간을 줄이기 위해서 Manchester carry chain을 사용하였다. 특히, Manchester carry chain에 Precharge를 사용하였으며, precharge를 사용하는 목적은 전압이 high에서 low로 떨어질 때에는 빨리 떨어지나 low에서 high로 올라가는 데에는 많은시간이 걸리므로 이것을 빨리 high 상태로 도달시킴으로서 지연시간을 줄일 수 있다.

carry-chain 회로는 그림 3-4(a)와 같다. 이것을 블록 다이어그램으로 나타낸 것이 그림 3-4(b)이다. Carry 발생회로를 세가지로 나누어 보면 다음과 같다.

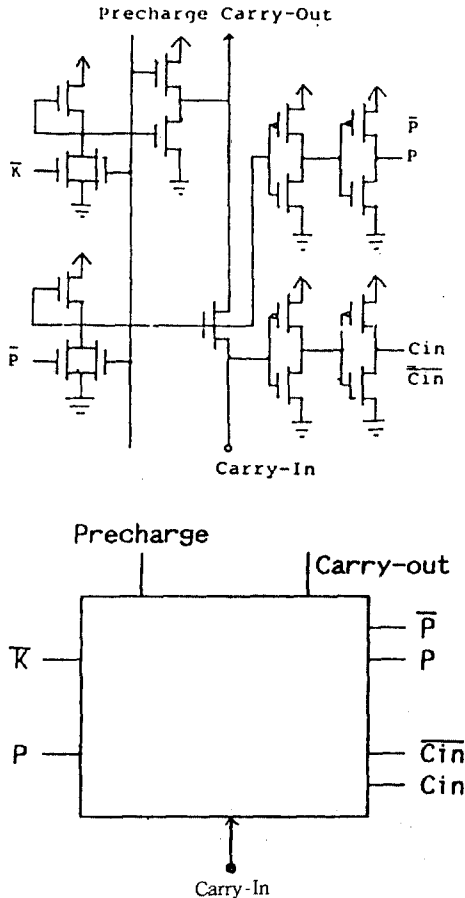


그림 3-4 Chain 회로도
Fig 3-4 Carry-chain circuit

첫째로, ALU의 null period 기간 동안에 pre-charge 에 의해서 캐리가 발생할 수도 있다. 둘째로, carry-kill 신호로부터 발생할 수 있다. 셋째로, carryin으로부터 발생할 수 있다.

III-2-3. ALU control driver

P, K, R 제어신호선은 그림 3-5와 같은 회로를 갖는다. 이 회로의 동작은 OP 코드 제어 신호 (0, 1)이 들어오면 ϕ_1 에 따라 제어되며 또한 출력 회로에 속도를 빠르게 하기 위하여 inverting super buffer를 사용하였다.

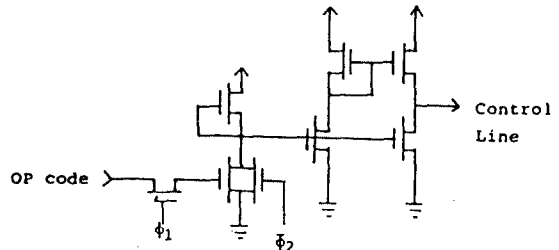


그림 3-5 ALU 제어 회로
Fig 3-5 ALU control driver circuit

III-2-4. 입력회로

입력회로의 입력은 ALU 입력(A, B), shifter, buses, 다른 source로부터 들어올 수 있다. 여러개의 입력중에 하나를 선택하는 것은 멀티플렉서에 의해서 즉 ($\Phi 1 \times \text{select}$)에 의해서 선택되며 통과 트랜지스터 $\Phi 2$ 에 의해서 내부에 보존된다. 회로는 그림 3-6과 같다.

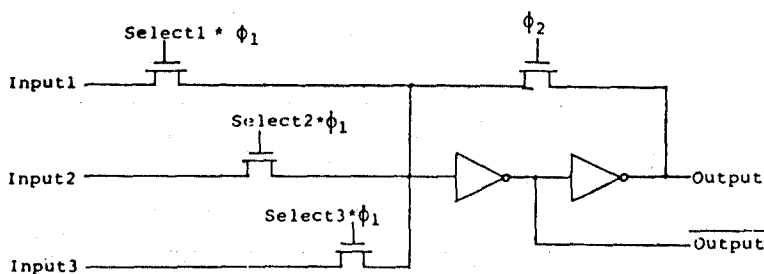


그림 3-6 입력 회로
Fig 3-6 Input circuit

III-2-5. 출력회로

출력 트랜지스터는 ALU 입력 트랜지스터와 비슷하나 타이밍이 다르다. $\Phi 2$ 끝에서 데이터를 가지게 되며 $\Phi 1$ 동안에는 내부에 보존. 출력 레지스터 회로는 그림 3-7과 같다.

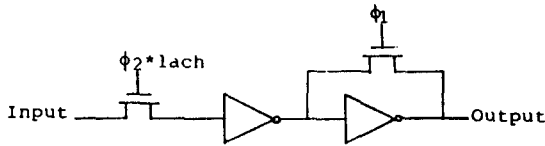


그림 3-7 출력 회로
Fig 3-7 Output circuit

III-3. BUS 회로

BUS의 구성은 BUS A와 BUS B의 두개의 BUS 로 구성되어 있으며 BUS는 BUS상에 있는 data를 $\Phi 1$ 동안에 전달한다. BUS 회로상에서 enable이 low이면 동작하지 못하고 high이면 동작한다. $\Phi 2$ 동안에 high voltage로 precharge 된다. 회로의 구성은 그림 3-8과 같다.

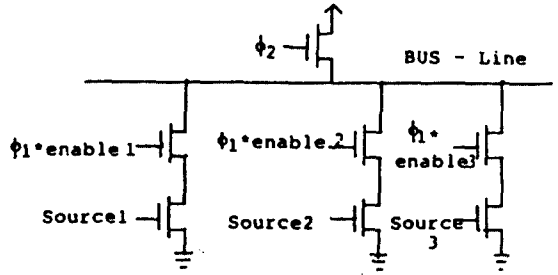


그림 3-8 BUS 회로
Fig 3-8 BUS circuit

III-4. Barrel shifter

Barrel shifter는 arithmetic 및 logic shifter rotate와 character insertion, extraction등과 같이 데이터를 정렬하는 것으로, 보통 extraction 기능을 강조하여 설계한 것을 funnel shifter라고 한다. shifting 블록은 그림 3-9에서와 같이 16×16 의 NMOS 트랜지스터 어레이로 되어 있어 원하는 bit(0~15) 만큼 입력 A와 입력 B를 shift할 수 있게 된다. 그림 3-10은 쉬프터(shifter)의 동작 결과를 나타낸다. Barrel 쉬프터의 동작은 쉬프터 정수에 의해서 결정된다.

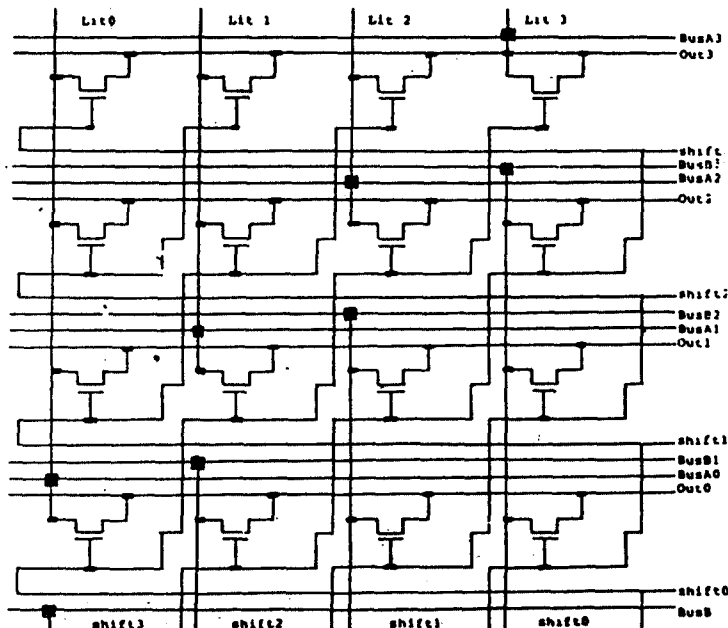


그림 3-9 Shifting 블록도
Fig 3-9 Shifting block diagram

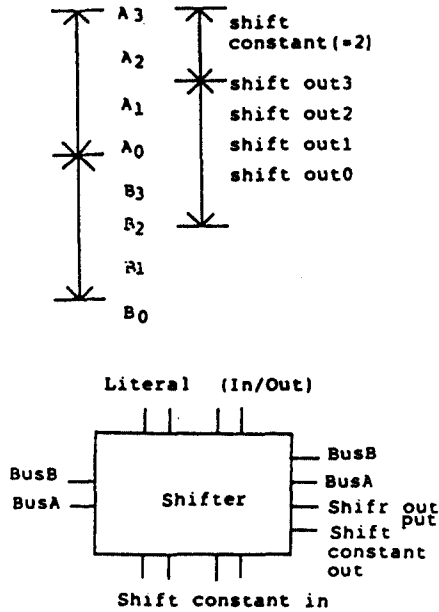


그림 3-10 Shifter의 동작 결과 및 블록도
Fig 3-10 Conceptual picture of the shifter's operatic

III-5, Register file

레지스터 화일은 명령의 operand 역할을 하는 것으로서 데이터 경로에서 사용하는 데이터를 메모리에서 레지스터로 이동시켜 전체적인 프로그램 수행 시간을 단축시키는데 목적이 있다. 본 논문에서는 16-bit 레지스터가 16개 있는 구조로 two-port로 구성되어 있다. 레지스터 화일의 구성은 two-port 레지스터 cell, 입력 멀티플렉서, 출력 드라이브, two-bus로 구성하였다.

III-5-1. Register Cell Array

레지스터 cell은 안정한 동작을 가능케 하며 two-port로 구성되어 있다.

회로의 구성은 그림 3-11과 같다.

III-5-2. Input, Output Driver

Data를 외부로 부터 받아들이거나 내부에서 외부로 내보낼 때 사용하며, 회로의 구성은 그림 3-12와 같다.

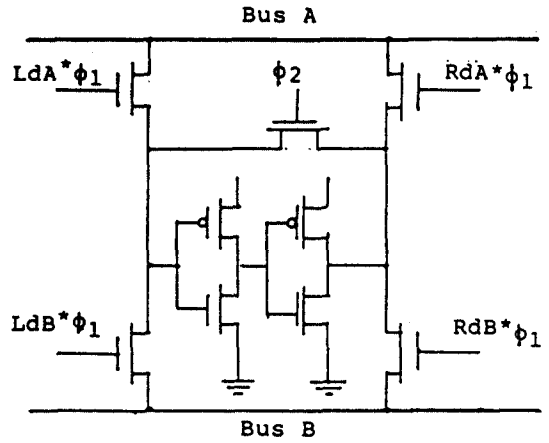


그림 3-11 2개의 portregister
Fig 3-11 two-port register cell

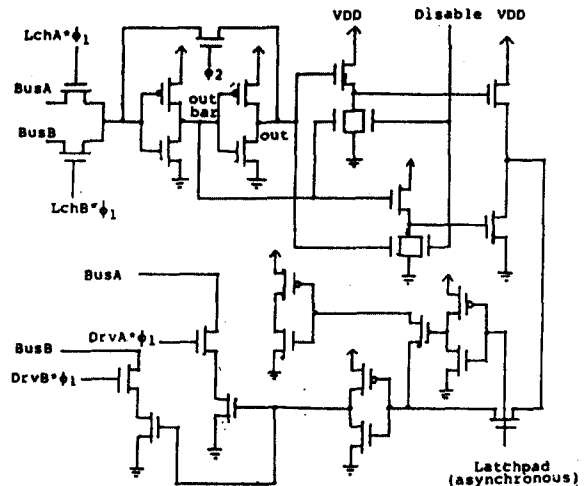


그림 3-12 Data port tri-state pad 회로
Fig 3-12 Data port tri-state pad circuit

IV. Simulation 및 결과

1. 칩의 여러개의 요소들은 두 개의 BUS에 의하여

연결되어 있고 $\Phi 1$ 클럭이 high인 기간 동안에 bus를 통해 데이터가 전달되고 클럭이 low 일때 BUS에 보존이 된다. 각 BUS는 어떠한 사이클 동안에 하나의 source로부터 데이터를 얻고 그리고 하나의 destination으로부터 데이터를 준다. left port, right port는 데이터 칩과 외부의 다른 소자에 연결되어 있다. Right port는 메모리 bus port를 left port는 시스템 bus를 가지고 있다. 각 port는 입력 latch, 출력 latch를 가지고 있으며 데이터의 전달은 다음 세가지 방법으로 전달된다.

- 1) $\Phi 1$ 동안에 BUS에서 받자마자 내보내는 것.
 - 2) $\Phi 2$ 동안에 데이터를 latch했다가 다음의 $\Phi 2$ 기간에 drive, 출력으로 내보내는 것.
 - 3) $\Phi 3$ 동안에 BUS에 데이터를 latch 시켰다가 핀 에이블이 충분히 low가 되었을 때 데이터를 출력으로 보내는 것.
2. 전체 시스템은 1 signal 2-phase 클럭을 사용하였다. $\Phi 1$ 이 high일 때 데이터 bit는 하나의 subsystem 으로부터 다른 subsystem으로 신호를 전달하고, $\Phi 2$ 가 high일 때 ALU가 $\Phi 2$ 기간 동안에 동작 한다. $\Phi 1$ 과 $\Phi 2$ 의 클럭 파형은 그림 4-1과 같고, ALU의 전체의 동작은 표 4 1과 같다.
3. 회로

기본 인버터의 이론을 토대로 하여 채널폭과 길이를 결정한 기본 논리 회로의 전기적 회로 특성을 조사하기 위하여 회로 시뮬레이터인 spice2를 사용하였다. 시뮬레이션에 사용한 spice2 파라미터는 2 μm CMOS poly 실리콘 게이트 제조 공정에서 추출한 파라미터 값이다. 그림 4-2 1 bit 전체의 회로도이다.

본 논문에서는 large circuit이므로 컴퓨터 용량이 한번에 모든 회로를 시뮬레이션 할 수가 없어서 몇 부분으로 나누어 시뮬레이션 하였다. 첫번째 부분은 입력 부분과 K, P, R 기능까지 시뮬레이션 하였으며, 두번째 부분은 캐리, R, out, bus회로까지 시뮬레이션 하였으며, 마지막으로 레지스터,

표 4-1 ALU의 동작
Table 4-1 Operation of ALU.

	K	P	R	Cin	Cond	
A+B	14	9	9	0	0	Add
A+B+Cin	14	9	9	1	0	Add with carry
A-B	13	6	9	2	0	Subtract
B-A	11	6	9	2	0	Subtract reverse
A-B-Cin	13	6	9	1	0	Subtract with borrow
B-A-Cin	11	6	9	1	0	Sbtract reverse with borrow
-A	3	12	9	2	0	Negative A
-B	5	10	9	2	0	Negative B
A+1	12	3	9	2	0	Increment A
B+1	10	5	6	2	0	Increment B
A-1	3	12	9	2	0	Decrement A
B-1	5	10	10	2	0	Decrement B
A B	15	6	6	0	0	Logical AND
A B	15	1	10	0	0	Logical OR
A B	15	9	10	0	0	Logical EXOR
A	15	12	10	0	0	Not A
B	15	10	10	0	0	Not B
A	15	3	10	0	0	A
B	15	5	10	0	0	B
Mul	14	1	14	0	0	Multiply step
Div	12	0	15	0	0	Divide step
A/O	15	1	10	0	0	Conditional AND/OR
Mask	5	10	8	0	0	Generate mask
SHLA	12	15	10	0	0	Shift A left
Zero	15	15	0	0	0	Zero

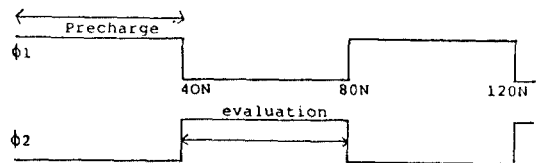


그림 4-1 $\Phi 1, \Phi 2$ 의 클럭 파형
Fig 4-1 $\Phi 1, \Phi 2$ clock phase

쉬프터를 시뮬레이션하였다. 그리고 K, P, R의 제어 기능에 의해서 원하는 동작을 구할 수가

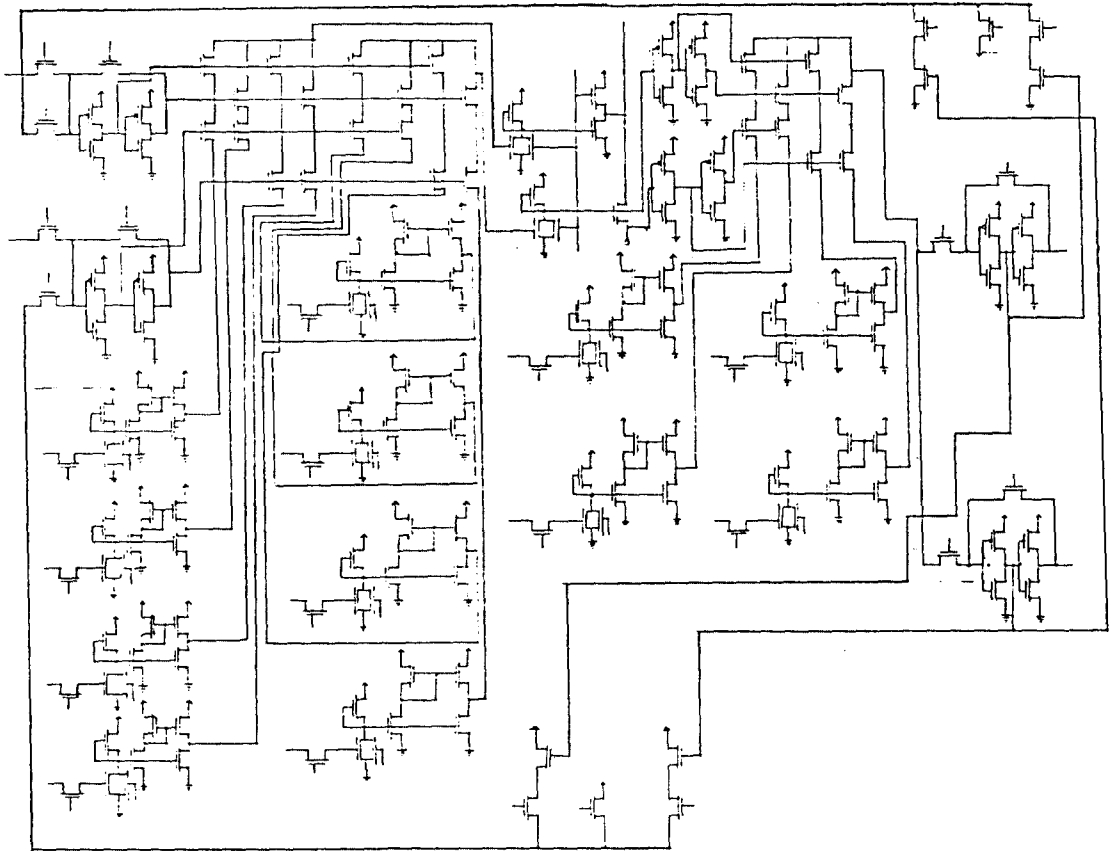


그림 4-2 1-bit 전체의 회로도
Fig 4-2 1-bit circuit of the whole.

있으며, K, P, R의 입력 및 회로의 조합에 의해서 K, P, R의 제어 값이 달라지며 이렇게 변하는 값들을 구할 수가 있다. ALU의 회로에서 가장 많은 시간이 소요되는 부분은 캐리 chine 부분이며 시간과 면적을 줄이기 위하여 맨체스터 회로에 NOR 회로와 precharge를 사용하였다. NOR 회로에서의 depletion mode의 채널 길이에 NOR의 동작이 좌우되었으며, 여러번의 회로 시뮬레이션에 의해서 최적의 채널값은 $12\mu\text{m}$ 를 얻어서 사용하였다. 캐리 chine은 첫단의 캐리 출력이 다음단의 캐리입력으로 하였으며, 구동능력을 위하여 super buffer를 사용하였다.

본 회로에서는 각기 4번째 stage마다 super buffer를 첨가하여 구동시켰으며, 16-bit Inverter Total 지연시간은 4.2 NS이며 carry chine에서의

지연시간은 4 NS이며 1 bit 당 Power 3mW, 16-bit 전체의 전력소모는 48mW이다.

결 론

본 논문에서는 고성능, 빠른 설계 시간 및 재 설계 기능성 부여 등에 주안점을 둔 단일 칩 16-bit $2\mu\text{m}$ 파라미터 공정에 의한 데이터 경로를 설계하여 회로 시뮬레이션하였다. 원칙적인 설계 방법의 체계적인 연구를 위하여 모듈화의 개념을 근간으로 한 설계 방법을 사용하였으며 이에 따라서 각 내부클럭이 bus에 연결되어 독립적으로 동작하는 subsystem이 되도록 하였고, 또한 명령의 추가가 용이하도록 시스템을 설계하였다. 데이

타 경로는 16-bit의 데이터를 처리하는 부분으로 산술 연산자, 레지스터 화일, barrel shifter 및 bus 회로로 구성되어 있다. 전체적인 시스템 설계에

선행하여 VLSI 시스템에서 공통적으로 자주 사용되는 기본 회로에 대한 분석 및 고찰을 하였다.

부 록 1

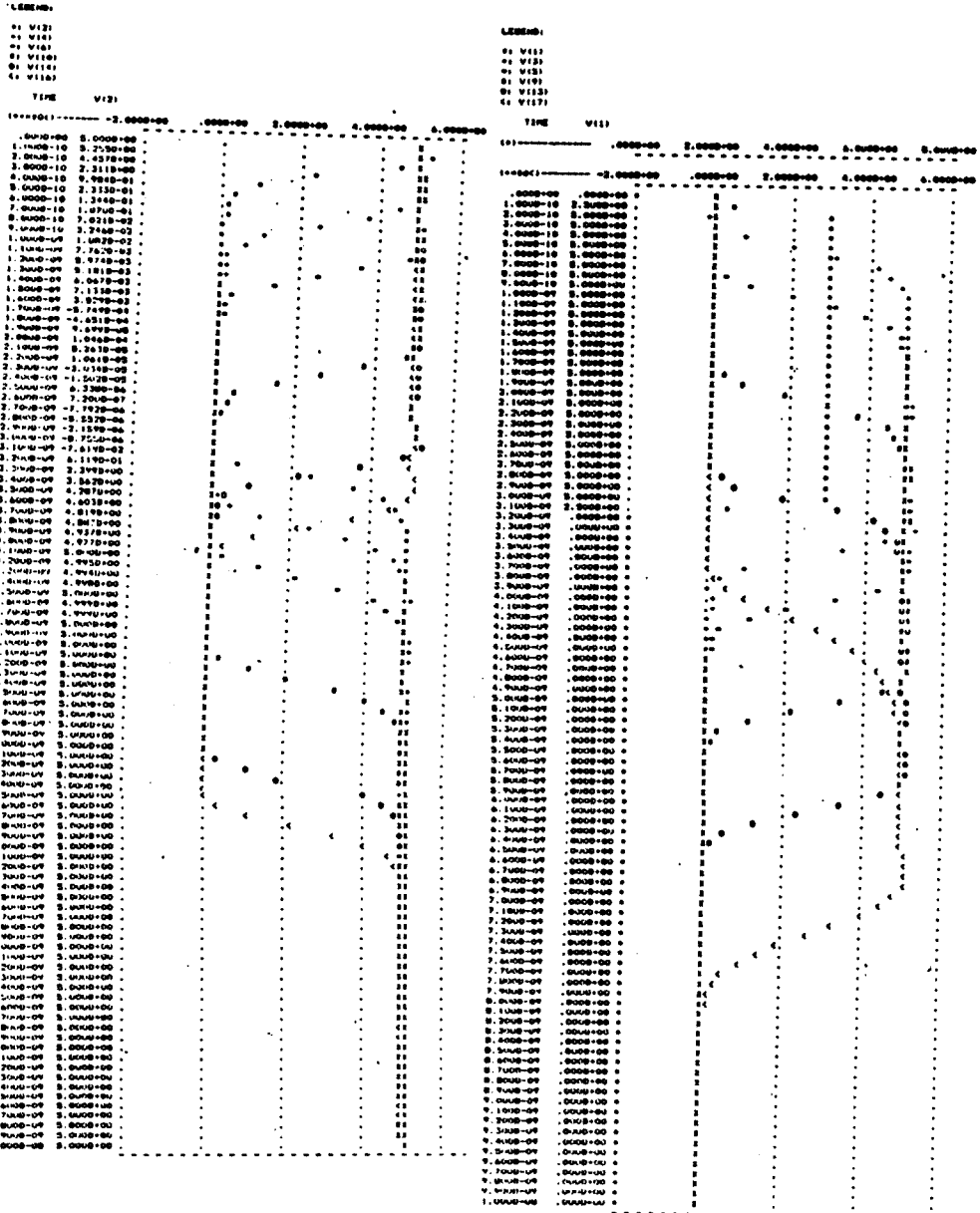
LEGEND: K·P·R gener logic funcion

- *: V(8)
- +: V(7)
- =: V(6)
- #: V(5)
- 0: V(9)

TIME	V(8)					
(**=#0)	-----	-5.000D-01	1.000D+00	2.500D+00	4.000D+00	5.500D
.000D+00	.000D+00	.	X	.	.	.
1.000D-09	.000D+00	.	X	.	.	.
2.000D-09	.000D+00	.	X	.	.	.
3.000D-09	.000D+00	.	X	.	.	.
4.000D-09	.000D+00	.	X	.	.	.
5.000D-09	.000D+00	.	X	.	.	.
6.000D-09	.000D+00	.	X	.	.	.
7.000D-09	.000D+00	.	X	.	.	.
8.000D-09	.000D+00	.	X	.	.	.
9.000D-09	.000D+00	.	X	.	.	.
1.000D-08	.000D+00	.	X	.	.	.
1.100D-08	.000D+00	.	X	.	0	#
1.200D-08	.000D+00	.	X	.	0.	#
1.300D-08	.000D+00	.	X	.	0	#
1.400D-08	.000D+00	.	X	.	.0	#
1.500D-08	.000D+00	.	X	.	.0	#
1.600D-08	.000D+00	.	X	.	.0	#
1.700D-08	.000D+00	.	X	.	.0	#
1.800D-08	.000D+00	.	X	.	.0	#
1.900D-08	.000D+00	.	X	.	.0	#
2.000D-08	.000D+00	.	X	.	.0	#
2.100D-08	.000D+00	.	X	.	0	X
2.200D-08	.000D+00	.	X	.	0	=
2.300D-08	.000D+00	.	X	.	.0	=
2.400D-08	.000D+00	.	X	.	.0	=
2.500D-08	.000D+00	.	X	.	.0	=
2.600D-08	.000D+00	.	X	.	.0	=
2.700D-08	.000D+00	.	X	.	.0	=
2.800D-08	.000D+00	.	X	.	.0	=
2.900D-08	.000D+00	.	X	.	.0	=
3.000D-08	.000D+00	.	X	.	.0	=
3.100D-08	.000D+00	.	X	.	.	X
3.200D-08	.000D+00	.	X	.	.	X
3.300D-08	.000D+00	.	X	.	.	X
3.400D-08	.000D+00	.	X	.	.	X
3.500D-08	.000D+00	.	X	.	.	X
3.600D-08	.000D+00	.	X	.	.	X
3.700D-08	.000D+00	.	X	.	.	X
3.800D-08	.000D+00	.	X	.	.	X
3.900D-08	.000D+00	.	X	.	.	X
4.000D-08	.000D+00	.	X	.	.	X
4.100D-08	.000D+00	.	X	.	.	X
4.200D-08	.000D+00	.	X	.	0	+
4.300D-08	.000D+00	.	X	.	0.	+
4.400D-08	.000D+00	.	X	.	0	+

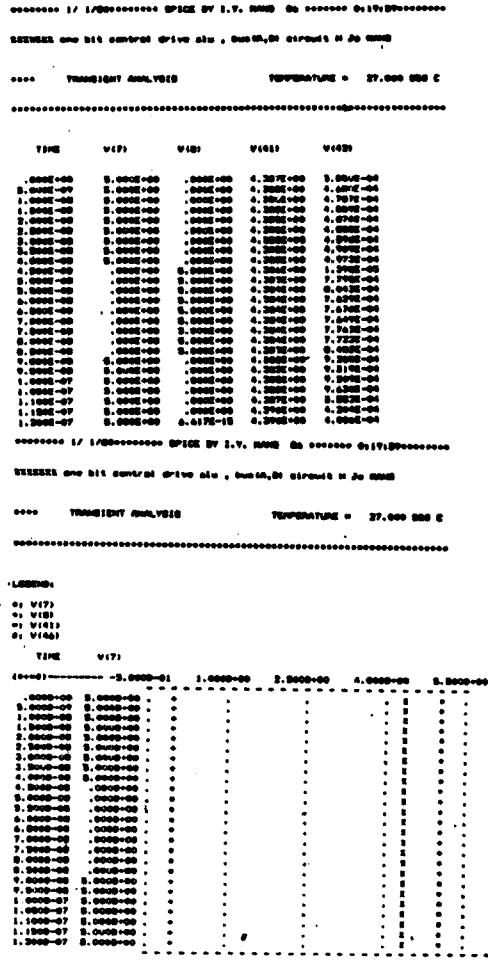
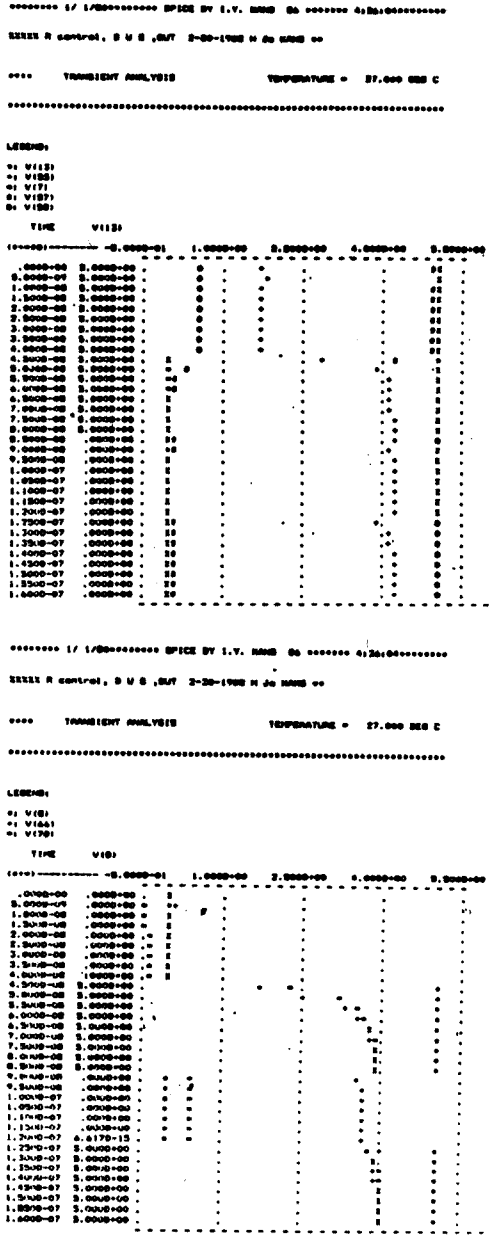
부 록 2

16-STAGE Inverter chain의 Output



부 록 4

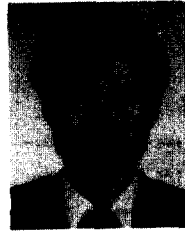
R, control, Bus, out의 output



JOB CONCLUDED
TOTAL JOB TIME 1074.36

參考文獻

1. Douglas V. Hall : Microprocessors and Digital System, Prentice-hall, Inc, 1986, pp. 345~350.
2. Saburo Muroga : VLSI SYSTEM DESIGN, Jone wiley & Sone, Inc., 1982 pp. 247~255, 411~413.
3. Neil H. E. Weste and K. Eshraghian : Principle of CMOS Design Addision Wesley, 1985 pp. 166~168, 310~326.
4. C. Mead and L. Conway, Introduction to VLSI system, 1980 Addsion-Wesley.
5. E. Horbst, Interdependence of architectue, Circuit design and technology, Advance, Course on VLSI, U. BRISTOL, 1982.
6. M. Morris. Mano : Digital Logic and Computer Design, Prentice-Hall, Inc 1979. pp. 385~401.
7. L. W. Nagel, SPICE2 : A COMPUTER Program to simulation semiconductor Circuit, Memo ERL-M520, University of California, Berklay, CA, May 9, 1975.



姜 熙 照(Heau Jo KANG) 正會員

1961年 1月26日生

1979年~1986年：圓光大學校電子工學科
卒業(工學士)

1986年~1988年：崇實大學校電子工學科
卒業(工學碩士)

1989年 8月~現在：韓國航空大學航空電
子工學科 博士課程

1988年 9月~1989年12月：중경工業專門
大學 講師

1988年 9月~1989年12月：大田實業專門大學講師

1989年 3月~1989年 8月：韓國航空大學 講師

1988年 9月~1989年12月：崇實大 電子計算院 講師

1990年 3月~現在：東新工科學 電子工學科 專任講師