

低電力 MCU core의 設計에 對해

안 형 근, 정 봉 영, 노 형 래
 삼성전자(주)

Abstract

With the advent of portable electronic systems, power consumption has recently become a major issue in circuit and system design. Furthermore, the sophisticated fabrication technology makes it possible to embed more functions and features in a VLSI chip, consequently calling for both higher performance and lower power to deal with the ever growing complexity of system algorithms than in the past. VLSI designers should cope with two conflicting constraints, high performance and low power, offering an optimum trade off of these constraints to meet requirements of systems.

Historically, VLSI designers have focused on performance improvement, and power dissipation was not a design criteria but an afterthought. This design paradigm should be changed, as power is emerging as the most critical design constraint. In VLSI design, low power design can be accomplished through many ways, for instance, process, circuit/logic design, architectural design, and etc..

In this paper, a few low power design examples, which have been used in 8 bit micro-controller core, and can be used also in 4/16/32 bit micro-controller cores, are presented in the areas of circuit, logic and architectural design.

We first propose a low power guidelines for

micro-controller design in SAMSUNG, and more detailed design examples are followed applying 4 specific design guidelines.

The 1st example shows the power reduction through reduction of number of state clocks per instruction. The 2nd example realized the power reduction by applying RISC(Reduced Instruction Set Computer) concept. The 3rd example is to optimize the algorithm for ALU(Arithmetic Logic Unit) to lower the power consumption. Lastly, circuit cells designed for low power are described.

약 어 표

AGU	: Address Generation Unit
ALU	: Arithmetic Logic Unit
Cin(·)	: Carry input
CK	: Clock
CPI	: Number of Clocks Per Instrucon
IPT	: Number of Instructions Per Task
db(·)	: Data RAM의 data bus
ib(·)	: CPU 의 internal data bus
Idec	: Instruction Decoder
log.	: Logic
MCU	: Micro Control Unit
MPEG	: Moving Picture Expert Group
MUX	: Multiplexer
PC	: Program Counter
PLA	: Programmable Logic Array
PROM	: Program ROM
reg.,H/W	: register, hard ware
RISC	: Reduced Instruction Set Computer
S(·)	: State Clock S1,S2,S3,S4
μ (·)	: 영역별 Micro Control Vector
μ (...)	: 전체 Micro control vector를 가리킴.

I. General Low Power Design Guidelines

당사에서는 저전력 MCU core를 만들기 위해 다음과 같은 design guideline을 사용하고 있다.^[8]

회로 및 신호 관련 Rule

Rule 1: Switching이 많은 signal에 대해 low capacitance routing을 사용한다.

Rule 2: 회로 signal에 의한 Glitch를 최소화한다.

Rule 3: Unsensitized signal이 早期 blocking 되게 설계한다.

Rule 4: Slew rate가 아주작은 signal이 발생되지 않게 설계한다.

Rule 5: Power소비가 작게 instruction을 encoding 한다.

Layout 관련 Rule

Rule 6: Non critical path에 대해 TR sizing을 최적화한다.

Rule 7: Buffer chain의 TR sizing을 최적화한다.

Rule 8: TR layout時 peak current가 작게 하여 EMI level을 줄인다.

회로 Cell 및 공정 관련 Rule

Rule 9: Standby mode가 많은 시간을 차지하는 경우 standby current를 줄일수 있게 reverse bias or multi-threshold 기술을 사용 한다.

Rule 10: Output pad buffer에 low threshold device를 사용하지 않는다.

Rule 11: 가능하면 TTL-compatible I/O를 사용하지 않는다.

Rule 12: 가능하면 dynamic logic 회로를 피하여 불필요한 precharged logic cell을 사용하지 않는다.

Rule 13: Low voltage process를 사용한다.

Rule 14: Cell mapping時 Pass Transistor Logic과 같은 저전력 cell을 사용한다.

CPU 구조관련 Rule

Rule 15: Chip內 여러 functional block이 동시에 동작 되지 않을때 power down mode가 가능하게 한다.

Rule 16: Input pin ordering에 의해 power 소비를 최소화 한다.

Rule 17: CPI를 줄이기 위해 state clock 數/cycle 를 최소화한다 .

Rule 18: IPT 가 크게 안늘어나는 범위에서 RISC like structure로 가서 control logic을 줄인다 .

Rule 19: MCU execution unit(EX: ALU)에 고속의 low power algorithm을 사용한다^[3].

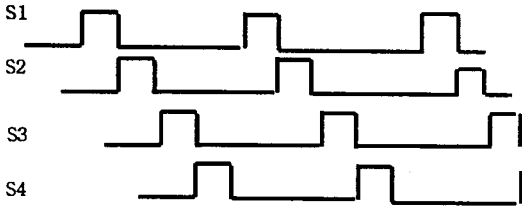
예를 들어 Rule 7을 설명하면 매우 큰 capacitive load구동時 buffer chain을 써서 delay를 최소화시켜야 과도전류가 적게흘러 low power화를 이룰수 있다.

다음장들에서 Rule 17, 18, 19, 14가 연속으로 설명된다. 향후 여러chip의 one chip 집적화가 가속되면서 chip size 축소와 저전력화는 더욱 더 중요한 과제로 부상될것이다.

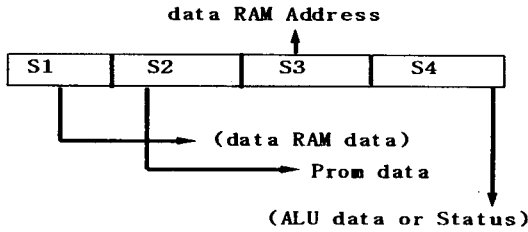
당사에서 특히 이런 설계 rule 들을 만든 것은 휴대형의 응용제품인 LCD, Pager, Data-bank 용 microcontroller들이 축전기 사용時 전력소모가 많으면 금방 새축전기로 교환해야 하는 문제가 발생하기 때문이다. 특히 저전력화되면 EMI noise도 줄게되고, 또 logic 회로량이 줄어들면 microcontroller의 chip size도 줄게되어 더욱좋다.

II. State CK數줄이기(Rule 17)

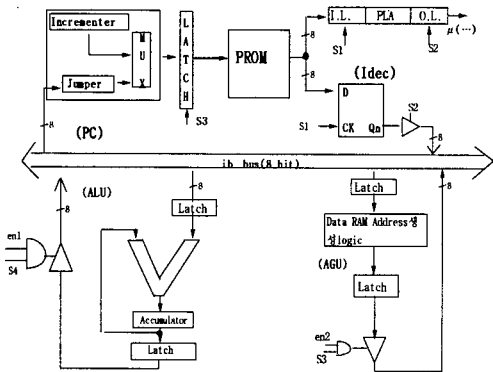
우리가 4state clock의 8 bit MCU를 그림 3과 같이 설계했다면, 이때 1byte instruction의 경우 그림 1과 같은 non overlapping 4 state clock이



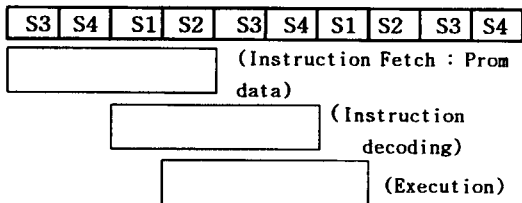
〈그림 1〉 4 state clock for 1 cycle timing圖



〈그림 2〉 MCU 내부 ib bus의 내용



〈그림 3〉 4 state CK MCU 핵심 파트 block 圖 *1.
I,O,L= In/Out Latch



〈그림 4〉 MCU Instruction Execution cycle 圖

instruction execution에 사용된다^[4].

그림 4에 MCU operation timing그림이 있다. S3에서 instruction fetch가 이뤄지고 (그림 3의

program counter value가 s3에서 latch된다.), micro control vector μ 를 생성하는 instruction decoding은 S1 clock 에서 이뤄진다(그림 3의 Idec IN buffer latch CK:S1).

일단 $\mu(\dots)$ vector가 생성되면 execution unit 가 동작하기 시작한다. $\mu(\dots)$ vector는 S2에서 생성 (그림 3의 Idec out latch CK:S2) 되어 execution은 S2에서 시작된다.

Instruction decoding에 S1,S2 2clock이 필요한 이유는 2 clocks이 필요한 이유는 multi byte instruction 또는 Skip조건 판정時 1 state clock 으로는 부족하기 때문이다.

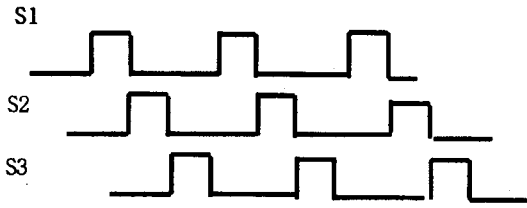
그림 2는 MCU core internal bus ib의 내부 내용을 나타낸다. 이는 그림 3의 모든MCU core의 ib bus로 向하는 buffer enable 신호로 결정된다. 모든 enable 신호가 disable되었을 때, peripheral 의 data RAM 의 data 가 ib bus에 실리기도 한다. 또한 2byte instruction의 경우 8 state clock 이 필요해 power소비의 증가가 나타나기도 한다.

여기서 불필요한 state clock을 없애고, power 소비를 줄이려면 그림 3을 참조하자. 여기서, program counter의 instruction fetch는 S3에서 이뤄지고, instruction decoding은 S1에서 시작된다. 이때 S4 state clock은 사실상 필요가 없다. 그림 3의 execution unit(ALU 또는 AGU)도 S2 (micro control vector $\mu(\dots)$ 생성시기) 이후 3 state clock동안 충분히 잘 동작한다.

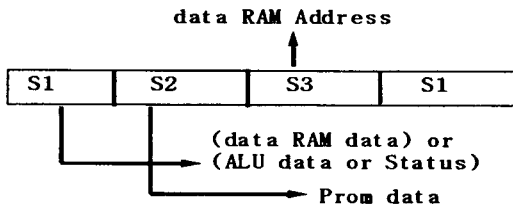
이렇게 S4 clock을 없애면 모든 unit의 시행시간이 3 state clock內로 이뤄지고, 또 시행도 1 state clock 빨라지게 된다.

그림 5,6,7에 3 state CK/ 1 cycle sequence圖가 있다. 그림 4는 3 state / 1 cycle system에선 S4 -> S1으로 되고 (S4없어짐), 그림 5는 S1, S2, S3 timing 圖 이다. 또한 그림 6는 ib bus내용을 보인다. 그림 2와 틀린점은 S4 state CK가 없어져 ALU시행결과가 S1 CK동안에 ib bus (internal bus)에 실리게 된다는 點이다.

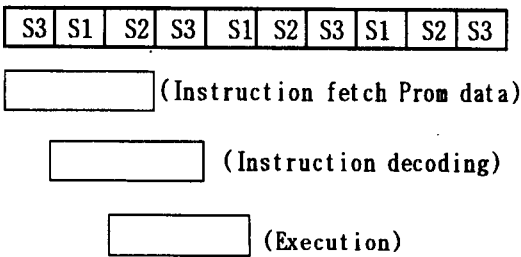
여기에서 data RAM의 data가 ib bus에 실릴 때는 ALU 결과가 ib bus로 출력되면 안되므로 (::ib bus data충돌) 이때는 ALU block의



〈그림 5〉 3 state clock for 1 cycle timing 圖



〈그림 6〉 ib bus 내용



〈그림 7〉 3 state CK MCU Instruction Execution cycle圖

enable 신호 en1은 0가 되어야 한다.

그림 7은 3 state CK/ 1 cycle MCU의 execution diagram이다. 그림에서 보듯 instruction fetch, decoding execution 모두 2 또는 3state CK內에 끝나게된다. AGU와 ALU 모두 micro control vector μ 가 생성되는 S2에서 다음 S2까지 3 state CK 동안에 충분히 원하는 동작을 수행할 수 있다. 이렇게하면 원래 MCU master CK=10 Mhz였다면 지금은 7.5Mhz 여도 되므로 power 소비는 25%줄게 된다^[6].

왜냐하면 power consumption은 동작 주파수에 비례하기 때문이다. RISC MCU와 같이 1 state clock / 1 cycle이면 최대 power가 75 %까지 준다^[1,3].

위에서, 4bit의 경우를 설명했지만, 8bit MCU의

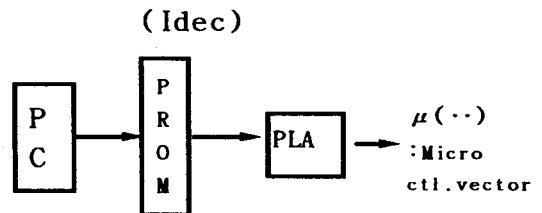
경우도 마찬가지이다. 특히 16/32 bit MCU의 경우 특정 TASK수행時 사용 algorithm에 따라 우리는 state CK數를 줄여 저전력화가 가능하다. 32bit MCU인 ARM7T RISC - Processor에서 그 multiplier H/W는 Booth algorithm을 쓴다. 이때 modified Booth algorithm을 쓰면 state CK數 / multiplying task 가 줄여 저전력화 할수있다.

또한 ARM7T MCU內의 thumb instruction decompressor(16 bit thumb instruction을 ARM instruction으로 변환)와 ARM instruction decoder의 instruction cycle time을 줄여서 state CK數를 줄일수 있다.

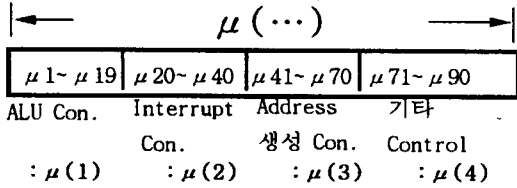
III. RISC like Structure(Rule 18)

원래의 RISC구조는 ①100개 이하의 instruction을 갖고 ②같은 길이의 instruction format을 갖고며 ③ 모든 instruction은 single cycle로 처리되고 ④ memory accessing이 load (store) 만으로 처리된다. 이렇게 할때 당연히 CPU control logic이 간단해져 고속, 低電力화가 이루어 질수있다. 이런 개념을 이용 당사 8bit MCU의 memory addressing中 indirect addressing을 없애고, 또한 instruction 中 bit manipulating instruction을 없애면 약 20%의 chip 크기 축소와 low power화를 이룰 수 있다.

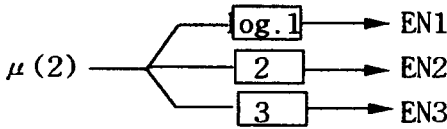
그림 8은 program counter로부터 나온 program ROM address에 의해 program ROM에서 instruction이 fetch 된후 PLA에 의해 micro control vector로 decoding 되는 신호 흐름圖이다.



〈그림 8〉 Instruction decoder(Idec)의 신호 흐름圖



〈그림 9〉 μ vector의 기능別 영역 分化



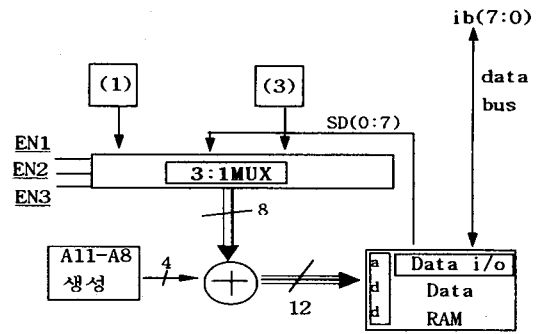
〈그림 10〉 Direct address enable 신호 EN1, Indirect address enable 신호 EN2, General register address enable 신호 EN3의 發生

이 micro control vector μ 는 그림 9에서 보듯 여러영역으로 나뉘어 MCU 각 block을 제어하게 된다. 예를 들면 $\mu_{41}\sim\mu_{70}(\mu(3))$ 는 address 생성 block (AGU block)으로 입력되어 그 block을 제어 하게 되는 것이다.

그림 9에서 μ vector는 90bit 로 구성되었다.

그림 10에 모든 addressing을 다 수행 할수 있는 memory address 생성 제어신호 logic이 있다. Logic1에 의해 EN1신호가 발생되고, logic2, logic3는 각 각 EN2, EN3 를 발생시킨다. 그림 11은 EN1,2,3에 의해 memory address가 생성 되는 AGU unit block 圖를 보인다. 그림 11에서 direct addressing일 경우 EN1이 선택되어(EN1=1, EN2= EN3= 0) (1)이 3:1 MUX 출력으로 나가고, MCU가 indirect addressing을 사용시 EN2가 선택되 SD bus 내용(HL register 내용)이 3:1 MUX 출력으로 나가 최종 A0-A7 address 가된다.

우리가 만약 여기서 indirect addressing을 안쓰는 구조로 MCU를 설계한다면 MCU의 AGU block圖는 그림 12와 같이 간단 해진다. 이런 RISC like 구조에선 없어진 instruction을 대신해 안없어진 instruction으로 똑같은 task를 수행하는 것은 가능하나 IPT가 커지거나, PROM size가 커질 수 있다. 하지만 이런점을 상쇄 하고도 더욱 저



* 註 : (1) Direct address A0-A7생성 logic
(3) Special register Address A0-A7 생성 logic

〈그림 11〉 원래의 MCU address 생성 block 圖

전력化 하므로 큰 문제는 없다.

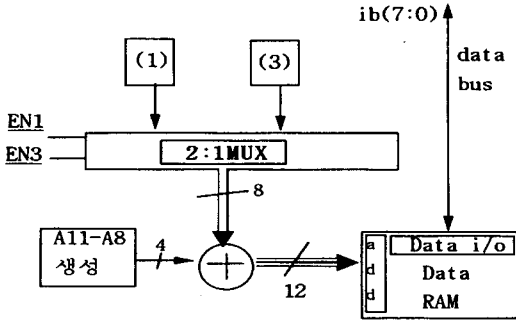
그림 11에서 EN3가 enable되면 (3) (special register address A0-A7)가 3:1 MUX출력으로 나가게된다. 이 register는 범용 MCU working register로 사용된다. 또한 A11-A8은 memory bank selection logic 으로서, memory bank 0에서 15 까지 구분한다. 또한 SD bus에는 data RAM 내의 working register中 하나인 pointer reg., HL register(indirect address pointer register) 內容 이 실려 EN2 가 enable될때 3:1 MUX 출력 으로 나와 최종 address A0-A7 이된다. 만약 B reg. 내용이 address 가되는 곳에 있는 내용을 accumulator로 옮기고 싶으면

```
LD HL, B; HL <- B
```

```
LD A, @HL; accumulator <- [ HL]
```

이 된다. indirect addressing이 없을때는 B reg. 내용을 계산또는 알아서 구한 다음, 그값을 direct address 로 하는 곳의 내용을 accumulator로 옮기면된다. 그 program은 LD A, DA; accumulator <- DA 번지의 내용 여기서 direct address DA는 B register 내용으로서 user가 어떤식으로든 그 값을 알아내야 한다.

그림 12에서 보듯 direct addressing만 사용하면 control logic의 경우엔 그림10의 EN2 신호 발생log. 2, 그림 11에선 SD bus가 필요 없어져, 3:



* 註: (1) Direct Address A0-A7 생성 logic
 (3) Special register Address A0-A7 생성 logic

〈그림 12〉 직접 Address only MCU AGU BLK 圖

1 MUX가 2:1 MUX로 간단해져 그림 12 처럼 된다. 따라서 chip size도 줄고 저전력도 이루어진다.

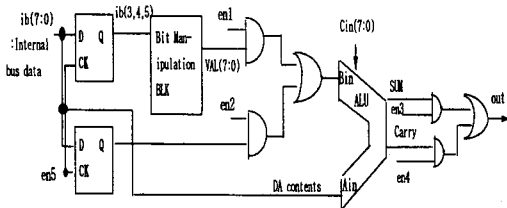
또 한가지 예로 ALU block을 살펴보면, 우리가 bit manipulation command를 다음과 같이 갖고 있다 하자.

LD DA, A; Accumulator A -> direct address DA

BSET DA.b; direct address DA 의 b번째 bit set 이 instruction을 수행하기위한 ALU H/W 가 그림 13 에 있다.

그런데 그림 9에서 ALU control신호는 $\mu(1)$ 이다.

$\mu(1)$ ($\mu_1, \mu_2, \dots, \mu_{20}$)에 의해 그림 13의 en1 = en 4=1, en2=en3=0, Carry in = Cin(7:0) = FFH면, ALU 두 8 bit 입력 A(7:0), B(7:0)에 대해 A는 direct address DA 의 내용이고 data RAM에서 internal bus ib(7:0)을 통해 ALU로



〈그림 13〉 Bit manipulation을 갖춘 ALU hardware(Adder 부분만보임)

ib(3, 4, 5)	VAL(7:0)	b	ib(3, 4, 5)	VAL(7:0)	b
000	00000001	0	100	00010000	4
001	00000010	1	101	00100000	5
010	00000100	2	110	01000000	6
011	00001000	3	111	10000000	7

〈그림 14〉 그림 13의 Bit manipulation block 진리표

입력된다. 한편 bit 정보는 b로서 program ROM 의 instruction에 저장되어 있고 그림 4 에서 보듯 S2에서 ib bus에 실려 ib(3,4,5)로 되어 그림 13의 bit manipulation block 으로 입력된다. 그 출력은 VAL(7:0)로서 en1=1에서 ALU의 B 입력이 된다.

VAL(7:0)는 bit 정보 decoding결과로서 b번째 bit만 set되었다. 한편 Cin=FFH, en3=0, en4=1에서 ALU out는 out= Carry out of Full adder = $Ain Bin + CinBin + Cin Ain = AinBin + Bin + Ain$ ($\because Cin=FFH \rightarrow 1111111b$) = $Ain + Bin = DA \text{ contents} + VAL(7:0)$ DA 내용중 b번째 bit이 set 된다. 그림 14는 bit manipulation block logic truth table 이다.

이부분 ALU를 그림 15와 같이 설계하면 bit manipulation은 불가능 해지나, 특정 bit을 set(예 b=3) 시키려면 다음과 같은 instruction으로 가능하다.

LD A,DA ; Acc <- DA

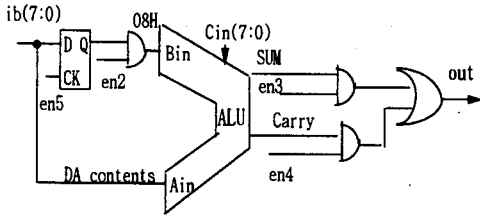
OR A, 08H ; 3rd bit set (if b=3)

LD DA,A ; Acc -> direct add DA

이럴경우, 특정bit set (여기서는 3rd bit) 시키기 위한 OR A,8H의 8은 user가 chip 밖에서 구해내야 한다.

즉 RISC like 구조는 쓰기가 불편해 지거나 IPT의 증가를 초래 할수 있다. 하지만 확실하게 chip size를 줄이고 또한 低電力化시키는데 아주 효율적이다.

그림 15의 경우 en5는 direct address DA내용의 3rd bit set를 위한 data 08H가 ib bus에 실렸을때 enable 되고 이것이 ALU의 B 입력으로 en2에 의해 들어간다.

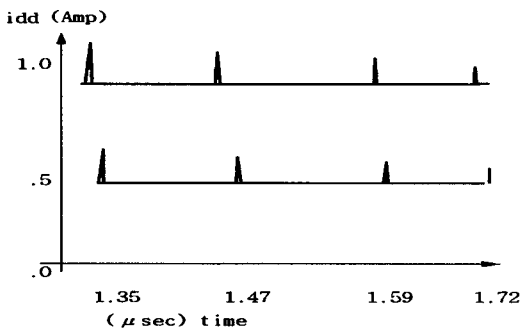


〈그림 15〉 Bit manipulation block 제거 시 New ALU block의 일부

ALU A 입력으로 Direct address DA내용이 ib bus를 통해 직접 연결된다.

그림 13 대비 그림 15는, bit manipulation block 없어지고 en1 신호 생성 logic도 없어져, low power 화가 이뤄진다. 그림 13의 회로와 그림 15의 회로를 각각 spice simulation 했을 때의 비교 결과가 그림 16에 있다.

여기서 입력조건은 direct address data rate 8 Mhz, en5 = 4 Mhz, en1=en4= 1, en2=en3= 0 로 했다. 단 그림 15에선 en2=1이다. 그림 16에서 보듯 RISC like 구조가 약 40% 이상의 idd (Vdd에서 chip으로 공급 되는 전류) 절감이 있어 저전력화 함을 알수 있었다. 그림 16은 당연히 0.125 μ s 주기 (8 Mhz)로 반복된다.



〈그림 16〉 그림 13 및 그림 15회로의 idd 비교

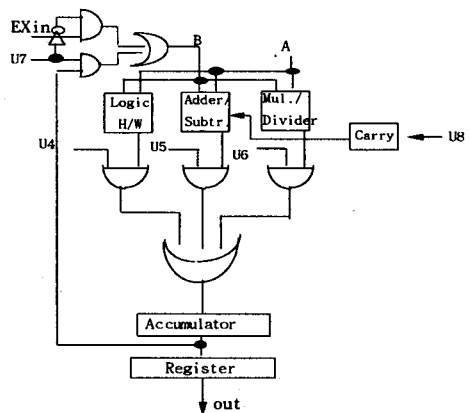
또한 CK주기(8 Mhz)로 전류가 peak값을 갖는다.

IV. ALU 저전력을 위한 최적algorithm 설계 (Rule 19)

ALU 설계時 최적 algorithm을 사용하면 저전력화 및 chip size 축소가 가능하다^[4,5]. 또한 두개의 execution unit가 同一 algorithm을 사용時, 1개의 H/W로 두 ALU function을 수행 해 역시 저전력화가 가능 하다. 전자의 경우 divider 설계時 전용 H/W를 쓰면 속도는 빠르나 H/W 회로 크기가 커져 전력소모가 많아진다. 따라서 adder와 multiplier를 써서 software algorithm으로 dividing을 하게된다. 이때 사용 algorithm이 최적화 안되면 program code density가 커져 속력이 느려지고 전력소모도 커지게된다. 후자를 설명하면 그림 17에서 우리는 일반적 ALU block 圖를 보인다.

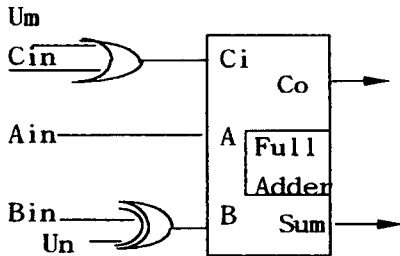
Ui는 micro control vector로서 예를들면, U4=1, U5=U6=0에서 ALU 출력은 logic H/W시행결과이고, U7=1에서는 ALU out이 다시 ALU의 B 입력 으로 들어가고, U7=0면 외부입력이 B로 연결된다.

Logic function에는 AND, OR, EXOR, NOT 등이 있다. 이때모든 logic function은 full adder의 carry, sum logic algorithm에 포함되 있으므로



〈그림 17〉 ALU block圖

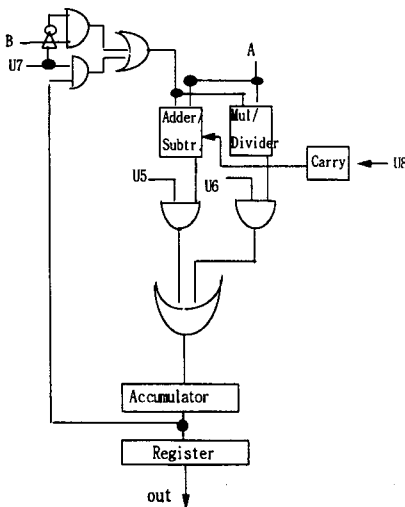
* Mul. : Multiplier, Subtr. : Subtractor, EXin : External input



〈그림 18〉 Full adder로 Subtraction施行

그림 17은 그림 19와 같이 logic 축약이 이루어져 저전력화된다. 또한 Subtraction이나, division도 각각 adding, multiplying algorithm에서 구현 가능해 같은 H/W를 쓴다. 예를들면 subtraction의 境遇 그림18에서 $U_n=1, U_m=1$ 을 넣어 $A+(B)+1 = A+2\text{'s complement of } B = A-B$ 로 뺄셈이, adder로부터 가능해진다.

그림 19은 17대비, logic H/W가 없어지고 3:1 MUX가 2:1 MUX로 변해 간단해 진다. 그림 20에 각각의 logic function이 full adder logic으로부터 구현되는 것을 보였다. 예를들면 self inverting은 Full adder(a,b,c 入力)의 Sum에서 $b=1, c=0$ 면 $S=a \oplus b \oplus c = a \oplus b = \bar{a}b + a\bar{b} = \bar{a}$ 로 a의 complementing이 실현된다. 우리는 8bit MCU의 ALU內 full adder의 algorithm을 이용



〈그림 19〉 저전력 ALU block 圖

S=Sum C=Carry out	Input to full adder(Ai, Bi, Ci입력)	Logic
C	$C_i=1$	$A_i \text{ (OR) } B_i$
C	$C_i=0$	$A_i \text{ (AND) } B_i$
C	$C_i=0, B_i=0$	\bar{A}_i
S	$C_i=0$	$A_i \oplus B_i$

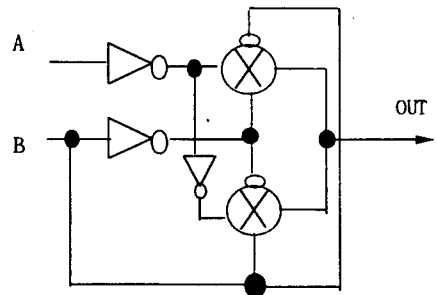
〈그림 20〉 Full adder(FA)로부터 logic function구현 table

logic function을 구현하여 저전력화를 꾀할수 있다.

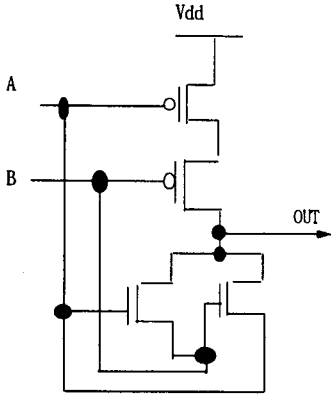
또한 그림 20에서 AND function은 full adder의 출력 carry out을 택하고 입력 carry in= $C_{in}=0$ 로 하면, $C_{out} = AB + BC_{in} + C_{in}A = AB + 0 = AB$ ($\because C_{in}=0$)로 AND function 구현이 가능하다. 이런 algorithm을 이용한 logic 축약은 여러 종류의 coder, decoder (EX: MPEG)에서도 찾아볼수 있어 coder로부터 decoder H/W를 쉽게 합성할수있다.

V. 저전력을 위한 Cell structure(Rule 14)

CPU structure가 결정이 되고 algorithm이 같을때, 우리는 사용된 cell의 저전력화를 통해 전체 micro-controller의 저전력화를 꾀할 수있다.^[2, 6] 예를 들면 EXNOR gate의 경우 그림 21과 같은 저전력cell을 만들수 있고, latch의 경우 D F/F

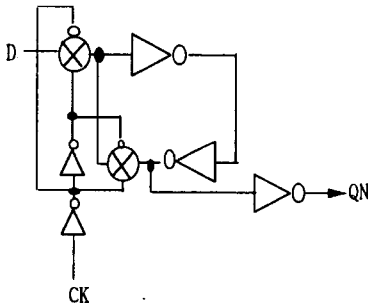


(a) 정규 EXNOR gate

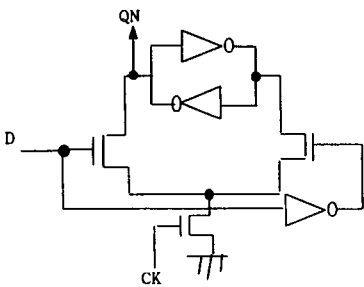


(b) 저전력 EXNOR gate

<그림 21> 정규 및 저전력 EXNOR gate



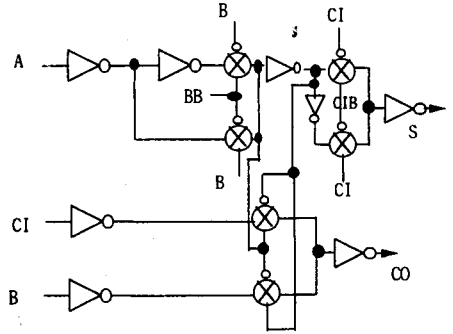
(a) 정규 level trigger D F/F



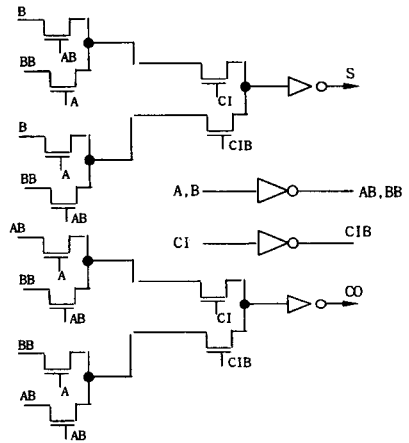
(b) 저전력 level trigger D F/F

<그림 22> 정규 및 저전력 level trigger D F/F

(JK F/F, SR F/F 도 마찬가지로)의 경우 그림 22와 같이 저전력화 된다. 또한 ALU 에서 add, subtract, multiply, divide에 항상 쓰이는 full adder의 경우 그림 23과 같은 CPL(Comple-



(a) 정규 Full adder S(SUM), CO(Carry Out)



(b) 저전력형 CPL 形 Full adder S(SUM), CO(Carry Out)

<그림 23> 저전력형 FA와 정규형 FA의 비교

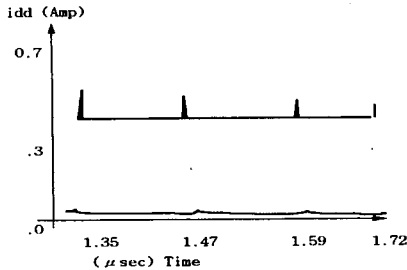
* CPL의 경우 저전력이면서, 고속이다.

mentary Pass Transistor Logic) type 저전력 cell을 사용할 수 있다.

그림 23은 1bit full adder로서, n bit의 경우 配線도 훨씬 CPL type가 간단하다. 이런 저전력형 cell에선 필요한 TR 개수도 훨씬 줄어 MCU chip size가 크게 줄어 든다. 그림 24에 저 전력형 D flip flop 과 CPL형 full adder 사용時, 그림 15의 RISC형 ALU의 add 과정변화를 보였다. 이것은 동일 입력에 對한 spice simulation 결과다.

정규cell 사용할 때보다 add(Power supply current)가 대폭줄어 저전력화함을 알수있다.

한편 EMI (Electro Magnetic Interference)의 경우도 저전력이 되면 add의 큰 감소에 따라 특성



〈그림 24〉 정규 cell과 저전력 cell사용시의 그림 15의 idd과형 변화

이 좋아진다(Low EMI emission).

VI. 결 론

이 논문에선 MCU의 低電力化를 위한 여러 설계 기법 들이 설명되었다. 당사에선 이런 기법들을 통합적으로 사용 기존 8bit MCU 對比 약 35%이상의 chip의 저전력화 및 크기축소를 얻을 수 있었다.

삼성전자 SYSTEM LSI본부의 향후 MCU core 사업 전개 방향은

- ① Module사업화 할 수 있게(MCU embedded 제품 가능하게) 모든 MCU core(4/8/16/32b core)를 ASIC의 digital core化한다.
- ② MCU core의 저전력화는 물론 MTP(Multi-Time Program) 가능하게 flash memory를 내장해 customer에게 제공한다.
- ③ 4bit의 저급 MCU 제품구조를 탈피 DSP 포함 8bit 이상의 고성능 MCU 제품을 설계 및 생산 Customer에게 제공한다.

등으로 요약할 수 있다. 당사는 이런 목적을 달성하여 세계제일의 MCU 제품회사가 되기 위해 부단히 노력 할 것이다.

참 고 문 헌

- [1] C. Piguet et al., "Low-Power Design of

8-b embedded CoolRisc Microm cores," IEEE J. of SSC, vol.32, No.7, pp.1067-1077, 1997.

- [2] Reto Zimmerman and W. Fichtner, "Low - power logic styles: CMOS vs. pass-transistor logic," IEEE J. of SSC, vol. 32, No 7, 1079-1086, 1997.
- [3] SEC 4b CMOS Microm(KS57 series), data book, Jan. 1997.
- [4] Anatha P chandrakasan et al., "Minimizing power consumption in Digital CMOSccts," proc. of IEEE., vol.83, pp. 498-523, April 1995.
- [5] Abdellatif Bellauar and Mohamed I. Elmasry, "Low power Digital VLSI Design", Kluwer Academic publishers, 1995.
- [6] Gerson A.S. Machado, "Low power HF Microelectronics", IEE circuits and systems series, 1996.
- [7] KAI HWANG, FAYE A. BRIGGS, Computer Architecture and Parallel Processing, pp177-181, McGRAW-HILL series in Computer, 1985.
- [8] 노형래, 정세웅, 이운태, 황경운, "Low Power Design Guidelines", 반도체 기술정보, 삼성전자, 1997. 8/9.

저 자 소 개



安 亨 根

1956년 11월 27일생, 1979년 2월 서울大 電氣科 학사, 1981년 2월 KAIST 석사(전기전자), 1988년 5월 美 New York 주립大(SUNY @SB) Ph.D.EE(전자박사), 1988년 5월~1998년 4월 현재 三星電子 部長(SYS. LSI 本部) 주관심 분야: MICOM, Telecom 용 LSI 설계 및 Marketing

저 자 소 개



鄭 璜 永

1955년 2월 9일생 1977년 3월 서울대학교 전자과, 1979년 8월 KAIST 석사 전자 및 전기과, 1992년 6월 박사(UCLA) 전기과, 1977년 3월~1998년 4월 현재 삼성전자 理事, 주관심 분야:MCU/

MPU 설계, Low Power회로 설계



盧 亨 來

1952년 9월 3일생, 1974년 2월 서울대학교 공과대학 전자공학과 졸업(학사), 1981년 12월 미 Rensselaer Polytechnic Institute, Computer Engineering(석사), 1996년 2월 고려대학교 전자공학

과(박사), 1982년 3월~1990년 1월 미 Intel사 VLSI 설계, Project leader, 1990년 1월~현재 삼성전자 LSI 사업부, LSI 사업부장