Journal of Historia Mathematica Vol. 9. No. 1, (1996), 12 - 31.

산술(算術)의 발전사(發展史) : 주판(珠板)과 컴퓨터[1]

아주대학교 정보 및 컴퓨터 공학부 **예 홍 진**

요 약

주판(珠板)은 지난 수천년동안 인류가 애용(愛用)했던 산판(算板, abacus)의 일종으로서, 휴대용 계산기(calculator)가 보편화되기 전까지 덧셈이나 곱셈 등의 사칙연산을 수행하는 데에 사용된 대표적인 산술도구(算術道具)이다. 한편, 자동화(自動化)를 위한 단순한 계산기계(計算機械)로부터 발전되어 온 컴퓨터는 오늘날 디지털 시대를 주도하면서 불과 수십년만에 우리의 삶과 생각을 완전히 새로운 모습으로 바꾸어 놓은 또 하나의 산술도구인 것이다.

본 논문에서는 주판에 적용된 여러 가지 산술원리들을 역사적으로 살펴본 뒤, 산술(算術)의 발전과 컴퓨터 사이에 어떠한 상관관계(相關關係)가 있는 지를 살펴보고자 한다. 이를 위하여, 새로운 산술원리를 컴퓨터에 도입하게 된 동기와 파급효과를 주판의 경우와 대비(對比)하여 설명하고, 현재 진행중인 컴퓨터 산술(computer arithmetic)분야의 연구동향을 토대로 미래의 컴퓨터를 전망한다.

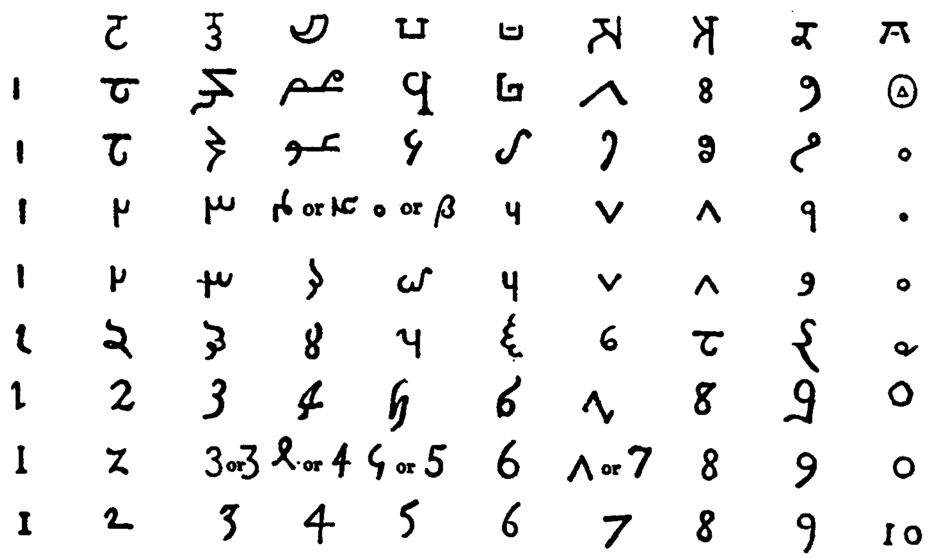
I. 서 론

언어(言語)와 문자(文字)를 사용함으로써 인류는 비로서 체계적이고 효율적인 상호간의 의사소통이 가능해졌으며, 이것은 문명(文明)이 탄생하게 된 직접적인 계기(契機)가 되었다. 그러나, 수(數)에 대한 개념이나 산술(算術)이 없었다면 과연 오늘날과 같은 과학기술의 비약적인 발전이 가능하였을까? 만일, 학교교육에서 수학(數學)이라는 학문영역이 얼마나 큰 비중을 차지하고 있는 지를 감안한다면 이러한 물음에 대한 해답은 명백해 보인다.

---

1) 이 논문은 1994년도 한국학술진흥재단의 공모과제 연구비에 의하여 일부 연구되었음.

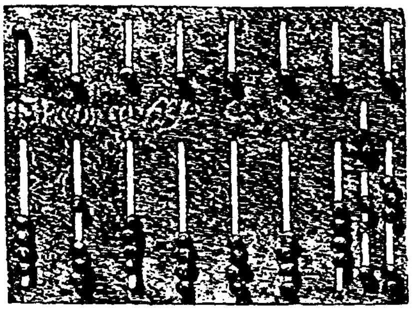
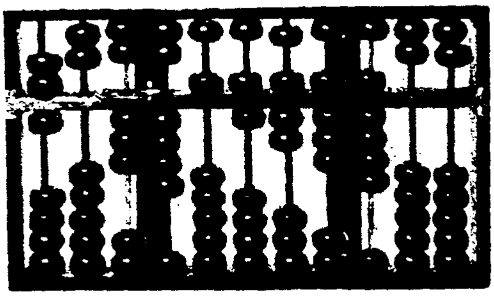
인간(人間)은 직접 구체적인 대상(對象)들을 다루면서 자연적으로 수(數)에 대한 개념을 가지게 되었으며, 점차 추상적이면서도 절대적(絶對的)인 수를 표현하기 위하여 기호(記號)를 사용하게 된 것이 숫자(數字)의 기원(紀元)으로 알려져 있다. 현재 전세계에서 널리 사용되고있는 아라비아 숫자는 원래 고대(古代) 인도사람들이 사용하던 숫자가 아라비아 상인들에 의하여 유럽에 전파되면서 널리 퍼지기 시작하였으며, 이러한 전파경로로 인하여 '아라비아 숫자'라는 이름이 붙여지게 되었다. [그림 1]은 고대 인도에서 사용되던 숫자들이 오랜 시간에 걸쳐 현재와 같은 모양으로 변화되는 과정을 보여주고 있다.



[그림 1] 아라비아 숫자의 변천과정([2] p.38)

사람들이 구체적인 물건의 개수를 헤아리던 것을 상대적으로 다루기 쉽고 휴대가 간편한 구슬이나 돌멩이, 조개껍질, 나뭇가지 따위의 작고 가벼운 물체들을 이용하여 수(數)를 표현하기 시작하면서 자연스럽게 원시적인 형태의 산판(算板, abacus)이 등장하게 되었다. 아직까지도 인류가 산판을 사용하기 시작한 것이 언제부터인지 분명한 기록은 없으나, 고고학계에서는 대략 5000년 전쯤으로 추정하고 있다.

고대(古代) 로마의 숫자표기나 중국의 숫자표기에는 자리수의 개념이 없었다. 예를 들어, '1708'을 로마숫자로 나타내면 'MDCCVIII'이며, 중국숫자로 나타내면 '一千七百八'과 같다. 따라서, 이와같은 숫자들만으로 계산하기가 쉽지 않았으므로, [그림 2]에서 보는 바와 같이 산판이나 주판(珠板)과 같은 보조도구를 사용하는 것이 보편적(普遍的)이었다. 로마의 산판은 정수(整數)뿐만 아니라 분수(分數)를 표현하는 데에도 사용되었지만, 여기에서는 분수에 대한 산술은 다루지 않기로 한다.

[그림 2] 고대 로마의 산판(左)과 중국의 주판(右)

아라비아 숫자가 유럽에 전파된 것은 단순히 새로운 형태의 숫자가 알려졌다는 사실보다는, 산술의 관점에서 볼 때 위치적 기수법(位置的 記數法, positional number system)에 의한 수체계가 도입되었다는 점에 더욱 큰 의미가 있다. 처음에는 작은 수들을 가지고 계산하던 것이 점차 사회규모가 커짐에 따라 보다 큰 수의 사용이 불가피하게 되었다. 새로운 수가 필요할 때마다 고유의 숫자와 이름을 붙이게 되면 그것들을 모두 기억하여 사용하기에는 한계가 있으므로, 같은 숫자라도 위치에 따라 실제 의미하는 수의 크기가 다른 표현방법이 필요하게 되었다. 이러한 위치적 기수법에 근거를 두고있는 아라비아 숫자가 유럽에 전해지면서 종이 위에서 계산하는 것이 기존의 산판을 이용한 방법보다 훨씬 쉬워진 것이다.

일반적으로 두 수를 더하거나 빼는 것과 같은 기본적인 사칙연산(四則演算)들을 처리함에 있어서, 위치적 기수법은 숫자를 써가면서 계산하는 필산(筆算)방식에 적합한 반면, 산판을 이용하여 계산하는 것은 수의 표현과정과 계산과정이 일치하므로 계산을 자동적(自動的)으로 처리하기에 적합하다. 이러한 차이로 말미암아 필산파(筆算派, algoristic school)와 산판파(算板派, abacistic school)간의 대립과 투쟁은 이미 중세(中世)시대부터 치열하게 진행되어 왔다.([2] pp. 167-169)

근대(近代)에 들어서면서 사람들은 종교의 속박을 벗어나기 위하여 노력했으며, 자유로운 사상과 학문에 대한 열망은 학술활동으로 이어져 과학과 기술이 급속하게 발전하게 되었다. 산업혁명으로 본격화된 대량생산(大量生産)에 따른 대량소비(大量消費)의 시대를 맞이하여 상업(商業)이 부흥하면서 산술에 대한 필요성이 사회적으로 크게 인식되기 시작하였다. 사회의 형태가 복잡해지고 재화(財貨)의 교환단위(交換單位)가 확대되면서 사람들은 이제 필산만으로 모든 계산을 처리하는 것이 벅차게 되었으며, 과거의 산판이나 주판보다 성능이 월등한 새로운 계산도구의 필요성을 갖게 되었다. 이것이 바로 컴퓨터 역사(歷史)의 출발점인 것이다.

계산하는 작업은 예나 지금이나 사람들이 가장 싫어하는 일중의 하나임에는 틀림없다. 초등학교 학생들이 가장 공부하기 어려워하는 교과목 중의 하나가 '산수(算數)'라는 통계자료는, 지나온 산술의 역사를 돌이켜보건대, 아마도 인간에게는 원래부터 산술능력이 충분히 주어지지 않았을지도 모른다는 의문을 갖게 한다. 여하튼, 복잡하고 골치아픈 계산으로부터 벗어나고픈 사람들의 욕망은 필산의 편리함을 위하여 고대(古代)에 사용되었던 수많은 숫자들 중에서도 결과적으로 아라비아 숫자를 선택하게 하였고, 나중에는 필산을 대체할 수 있는

계산기계를 발명케 한 원동력이 된 셈이다.

다음 장에서는 산술의 역사속에서 산판을 몰아내고 필산을 주도했던 10진법이 컴퓨터의 등장으로 인하여 2진법에 자리를 내주면서 디지털 시대가 활짝 꽃피우게 된 배경을 설명하고 있다. 3장에서는 덧셈과 곱셈이 주판에서 처리되는 과정을 컴퓨터의 산술처리부분과 비교하여 설명한 뒤, 컴퓨터 산술의 발전과정을 역사적으로 정리한다. 끝으로 4장에서는 컴퓨터에 전기(電氣)대신에 빛(light)이나 레이저(laser)와 같은 새로운 매체를 동력원(動力源)으로 사용함으로써 10진법이나 2진법이 아닌 새로운 수체계의 도입 가능성을 검토하고, 그에 따른 산술의 개혁(改革)이 만들어 낼 새로운 컴퓨터의 모습을 미리 살펴보기로 한다.

Ⅱ. 10진법에서 2진법으로의 전환(轉換)

수(數)를 표현하는 방법을 크게 나누어 볼 때, 숫자를 사용하여 종이에 쓰는 방법(기수법, 記數法)과 각각의 수에 이름을 붙여 부르는 방법(명수법, 命數法)이 있다. 이러한 두 가지 방법은 쓰여진 수를 읽거나 불리워진 수를 기록하는 과정에서 서로 밀접한 관계를 가지고 있다. 예를 들면, 우리 나라에서는 일, 이, 삼, 사, ... 등과 하나, 둘, 셋, 넷, ... 등의 두 가지 수사(數詞)를 사용하고 있으며, '삼십칠'이나 '서른일곱'이라는 말에 '37'이라는 수를 종이 위에 기록한다. 이러한 사회적 약속은 일상생활에서 수를 쉽게 표현할 수 있도록 고안되었으며, 사회의 규모가 커지면서 더욱 큰 수를 표현하기 위하여 수체계(數體系)를 확장하게 되었다.

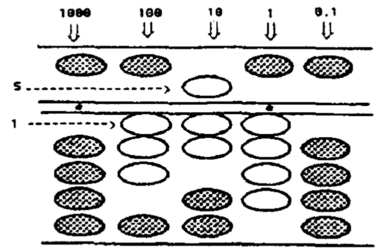
한편, 언어나 숫자에 의존하지 않고 신체의 일부분을 이용하여 수를 표현하는 방법도 사용되었다. 손과 발은 항상 휴대(?)할 수 있는 가장 대표적인 보조도구로서, 손가락과 발가락의 개수에 따라 5, 10, 20과 같은 수를 기본단위로 사용하는 것은 아주 자연스러운 발상(發想)인 것이다. 로마시대의 기수법을 보면 Ⅰ, Ⅱ, ⋯, Ⅴ, Ⅵ, ⋯, Ⅹ, Ⅺ, ⋯, XV, ⋯ 등으로 5진법을 사용한 흔적이 엿보인다. 아즈텍인들의 명수법에는 1, 2, 3, 4, 5, 10, 20, 40 ⋯ 에 대해서 특별한 이름이 존재하였으며, 마야인들은 0은 물론 1부터 19까지의 기호를 정의하여 20, $18\times20=360$, $18\times20^2=7200$, ⋯ 등을 기준수(基準數 : key number)로 정한 위치적 기수법을 사용하였다.([2] pp. 42-43) 고대 중국에서는 손가락을 좀더 세분하여 엄지를 제외한 나머지 네 손가락의 마디 마디의 개수를 모두 합한 12를 기본단위로 사용하여, 10간(干)과 12지(支)로 이루어진 갑자(甲子), 을축(乙丑), 병인(丙寅), ⋯ 등의 60간지(干支)를 손가락 마디마디를 따라 헤아릴 수 있었다. 또한, 고대 바빌로니아에서 슈메르인들이 사용했던 쐐기형 문자(cuneiform characters)로 쓰여진 진흙판들에 새겨진 수학적 내용들을 해독(解讀)한 결과, 놀랍게도 그들이 수를 표현함에 있어서 60을 기본단위로 사용했음이 밝혀졌다.([2] p.30)

이러한 기본단위를 정하여 수를 표현하는 것을 진법(進法)이라 부르며, 기본단위를 진수(進數)라 한다. 인류는 수(數)를 사용하기 시작하면서 각 시대의 변천(變遷)에 따라 여러 가지 형태의 진법들을 사용하여 왔다. 프랑스 사람들은 기수법(記數法)으로 10진법을 일관되게 사용하고 있음에도 불구하고, 명수법(命數法)은 현재에도 10진법과 20진법 그리고 60진법을

혼용하고 있다. 예를 들면, 100보다 작은 정수(整數)에 대하여 1부터 69까지는 10진법에 따라 un(1), deux(2), trois(3), …, soixante-neuf(69 = 60 + 9)로 부르는 반면, 70부터 79까지는 60진법에 따라 soixante-dix(70 = 60 + 10), soixante-onze(71 = 60 + 11), …, soixante-dix-neuf(79 = 60 + 19)로, 80부터 99까지는 20진법에 따라 quatre-vingts (80 = 4 × 20), quatre-vingt-un(81 = 4 × 20 + 1), …, quatre-vingt-dix-nenf(99 =4 × 20 + 19)로 부르고 있다. 그렇다면, 왜 인류는 10진법을 최종적으로 선택하게 되었는가? 이 질문에 대한 답은 아직도 분명치 않다. 다만, 10진법에 근거한 아라비아 숫자들이 유럽에 전파되어 필산(筆算)에 편리하게 사용되면서, 이것을 대체할 만큼 우수한 다른 수체계를 발견하지 못했거나 혹은 새로운 진법의 필요성을 느끼지 못했기 때문이 아닌가 추측할 뿐이다.

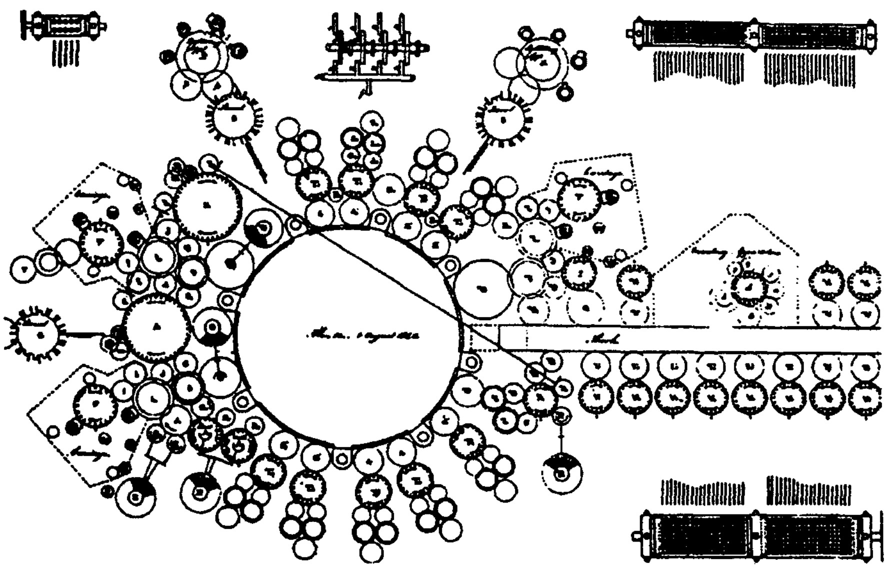
명수법(命數法)과 기수법(記數法)에서 10을 진수로 사용하는 것은 산판의 형태와 이용방법에도 영향을 미치게 되었다. 소위 산판이라고 불리는 것은 어느 시대에서나 제가끔 다양한 형태로 널리 사용되어 왔다. 모든 산판들은 공통적으로 평면을 구분하여 그 구간내에 작고 가벼운 물체들을 올려놓거나 특정 기호를 표시함으로써 여러 가지 다른 수를 표현하고 있다. 10진법이 도입되면서 위치적 기수법의 특성에 따라 산판의 구간들에는 차례로 1, 10, 100, 1000, … 등과 같은 10의 거듭제곱수가 각각의 자리수로 정해지게 되었다. 예를 들어, 44의 경우, 앞의 '4'는 10의 자리수에 있으므로 40을 의미하고, 뒤의 '4'는 1의 자리수에 있으므로 4를 의미한다. 또한, 309와 같이 10의 자리수에 해당되는 내용이 없는 것을 표시하기 위하여 숫자 '0'(零, zero)이 사용되었다. 그러나, 우리가 지금 사용하는 것과 같이 항상 왼쪽부터 오른쪽 방향으로 가장 높은 자리수부터 내림차순으로 정의한 것은 아니며, 한자(漢字)나 아라비아 문자처럼 위쪽에서 아래쪽 방향으로 또는 오른쪽에서 왼쪽 방향으로 차례대로 정의하는 경우도 있었다.

우리 주위에서 아직도 흔히 볼 수 있는 주판(珠板)에 대하여 살펴보기로 하자. 현재 우리가 사용하고 있는 주판은 본래 중국의 주판이 다양한 형태로 변형되어 사용되다가, 근대(近代) 이후에 일본에서 개량된 것으로 알려져 있다. 주판에는 자리수의 혼동을 방지하기 위하여 4칸마다 일정하게 점이 찍혀 있는데, 이것은 일(一), 십(十), 백(百), 천(千), 만(萬), 십만(十萬), 백만(百萬), 천만(千萬), 억(億), … 등의 명수법에 따라 일, 십, 백, 천이 반복적으로 사용되었기 때문이다. 0부터 9까지의 숫자를 표현함에 있어서는 각각의 자리수 구간을 다시 상하(上下)로 구분하여 위쪽은 5의 자리수를 아랫쪽은 1의 자리수로 정하였다. 이러한 변형은 많은 구슬의 이동에 따른 번거로움을 피하고, 손가락 두 개-엄지와 검지를 동시에 사용함으로써 효과적으로 수를 정확하게 표현하기 위한 것이다. [그림 3]은 주판에 374라는 수가 표현된 모습을 보여주고 있다. 이와 같이, 오늘날의 주판은 10진법과 5진법을 혼용하여 손가락의 움직임이 빠르고 편리하도록 고안된 것이며, 주판을 이용한 산술에 대해서는 다음 장에서 다루기로 한다.



[그림 3] 주판(珠板)을 이용한 수(數)의 표현 예(例)

동양(東洋)에서는 한자문화권(漢字文化圈)의 영향으로 근대(近代)에 이르기까지 주판이 널리 보급된 반면, 서양(西洋)에서는 아라비아 숫자의 도입으로 산판이 자취를 감추고 필산이 주로 사용되었다. 17세기에 들어서면서 톱니바퀴를 이용한 수의 표현방법이 활발하게 연구되면서, 주판보다 성능이 뛰어나고 자동화된 계산도구들이 잇따라 발명되기 시작하였다. 톱니바퀴의 회전에 의하여 덧셈이 자동적으로 계산되는 동작원리는 사람에 의하여 수의 표현과 계산이 전적으로 이루어지던 이전의 산판이나 주판과는 전혀 다른 획기적인 발명이었다. 이러한 의미에서 '세계 최초의 계산기'라 일컫는 파스칼(B. Pascal)의 계산기가 등장한 이후, 라이프니츠(G. Leibnitz)의 가감승제가 가능한 계산기, 배비지(C. Babbage)의 대수표 계산을 위한 미분기와 [그림 4]와 같은 미완성 계산기에 이르기까지 수세기동안 톱니바퀴의 원리를 응용하여 빠르면서도 정확하고 사용하기에 편리한 계산기를 만들려는 노력은 계속되어 왔다.



[그림 4] 배비지(C. Babbage)의 미완성 계산기 설계도면([8] 표지)

'Analytical Engine'이란 이름이 붙여진 배비지의 미완성 계산기는 실제 제작되어 사용되지는 못했으나, 하나의 장치로 여러 가지 서로 다른 연산들을 처리할 수 있도록 고안되었다는 면에서 오늘날의 컴퓨터 프로그램과 유사한 기능을 가진 것으로 평가되고 있다. 이러한 그의 생각은 놀랍게도 현재 우리가 사용하는 컴퓨터의 기본구성요소인 입·출력장치, 기억장치, 연산장치 그리고 제어장치 등을 적어도 개념적으로는 모두 갖춘 것이다.

특히, 잭쿼드(J.-M. Jacquard)가 발명한 방직기가 천공카드를 이용하여 다양한 무늬의 직물을 생산하는 점에 착안하여, 최초로 천공카드방식의 입력장치를 채택한 것은 주목할 만하다. 이러한 그의 아이디어는 반세기가 넘게 지난 뒤에야 홀러리스(H. Hollerith)가 제작한 'Census Machine'에 반영되어 널리 사용되기 시작하였다. 천공카드를 이용한 입력장치의 개발은 2진법에 따른 수의 표현방식을 계산기계에 적용한 최초의 시도로서, 나중에 부울(G. Boole)이 논리대수(logical algebra)라는 이론적 기반을 세우고, 테슬러(N. Tesla)가 게이트(gate) 또는 스위치(switch)라고 부르는 논리회로(logical circuit)를 발명하게 됨으로써 디지틀시대를 맞이하게 되는 결정적인 열쇠가 되었다.

결국, 배비지가 설계했던 'Analytical Engine'이라는 계산기계는 에이큰(H.H. Aiken)에 의하여 진공관과 전자식 계전기(繼電器)로 구성된 MARK-I이라는 거대한 자동계산기가 만들어짐으로써 무려 100여년의 시간이 흐른 뒤에야 현실적으로 구현될 수 있었다. ABC(Atanasoff-Berry Computer), MARK-I, EDVAC(Electronic Discrete Variable Automatic Computer), EDSAC(Electronic Delay Storage Automatic Computer), UNIVAC-I(Universal Automatic Computer-I) 등과 함께 제 1 세대(世代) 컴퓨터들을 대표하는 ENIAC(Electronic Numerical Integrator And Calculator)은 진공관을 기억소자(記憶素子)로 사용하였고, 비록 10진법 수체계를 그대로 사용하였지만 내부적인 계산을 위하여 2진연산(binary operation)을 도입하였으며, 폰노이만(J. von Neumann)이 제안한 프로그램 내장(內裝)방식을 채택하였다는 점에서 최초의 전자식 계산기라 불려지게 되었다.

지속적인 전자기술의 발달로 진공관을 트랜지스터가 대체하고 IC(Integrated Circuit), LSI(Large-Scale Integration), VLSI(Very-Large-Scale Integration) 등과 같은 새로운 기술이 잇따라 개발되면서 제 2 세대, 제 3 세대, 제 4 세대 컴퓨터들이 등장하게 되었다. 이러한 과정에서 문자와 숫자는 물론 음성, 사진, 오디오, 비디오 등과 같은 다양한 형태의 문화자료(文化資料)들이 비트(bit)라는 2진소자(素子)들에 의하여 표현이 가능해지면서 숫자화(數字化, digitized)되고, 본래 산술을 위한 2진연산은 각종 정보처리(情報處理)에 응용(應用)되었다.

하나의 톱니바퀴가 회전하면서 다음 자리수의 톱니바퀴를 한 칸씩 움직이는 기계식 계산기의 등장으로 사람들은 산술을 수(數)의 논리적인 계산방식(計算方式)으로 인식(認識)하기보다는 기계운동에 의한 자동적(自動的)인 현상으로 받아들이게 되었다. 처음에는 10진법에 따라 0부터 9까지의 숫자가 씌여진 10개의 톱니를 가진 톱니바퀴가 사용되었으나, 계산기의 부피를 줄이고 동력전달을 원활하게 하기 위하여 톱니의 개수를 줄이면서 사람들은 10진법이 아닌 다른 진법의 효용가치에 관심을 가지기 시작하였다. 회전운동과 상하왕복 운동 사이에 기계적인 변환은 천공 카드방식의 도입에 따라 10진법과 2진법간의 수의 변환에 그대로 적용될 수 있었다.

한편, 다양한 산술기능을 갖추기 위하여 엄청나게 많은 톱니바퀴들이 서로 복잡하게 연결

되었으며, 이러한 거대한 기계를 실제 움직이기 위한 동력(動力)을 얻기란 주로 소나 말등의 가축의 힘에 전적으로 의존하던 당시(當時)로서는 불가능한 것이었다. 이러한 관점에서 볼 때, 진공관의 발명은 톱니바퀴의 회전운동을 전자의 흐름으로 대체시켜 줌으로써 전기(電氣)라는 에너지원(源)을 계산기계의 동력원으로 사용하게 된 직접적인 계기가 되었다. 반도체(半導體)의 개발과 제작기술의 발전은 곧 전자식 계산기의 발달로 이어져, 2진법에서의 산술은 필산(筆算)이나 암산(暗算)을 위한 것이 아니라 효율적인 논리회로 설계를 위한 하나의 도구로서 새로운 연구대상이 되었다. 이로써 '컴퓨터 산술(Computer arithmetic)'이라는 새로운 학문분야가 생겨난 것이다.

전자식 계산기(Calculator)에서 시작된 컴퓨터(Computer)의 발달은 10진법에서 2진법으로의 전환(轉換)을 고착화시켰으며, 필산(筆算)을 위한 인간중심(人間中心)의 산술(算術)이 컴퓨터 내부의 연산처리를 위한 기계중심(機械中心)의 산술로 바뀌게 되었다. 수천년동안 다양한 수체계가 만들어지고 변화하고 사라지는 과정속에서 필산의 용이함으로 인하여 가장 널리 보급된 10진법이 컴퓨터의 등장(登場)에 따라 불과 수십년만에 인류가 최초로 사용했던 가장 원시적이고 단순한 수체계 중의 하나인 2진법으로 되돌아 간 것은 역사(歷史)의 아이러니가 아닐 수 없다.

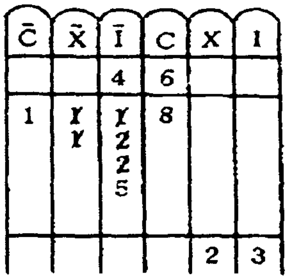
Ⅲ. 산술(算術)의 발전과정(發展過程)

인간이 고안한 최초의 자동화된 산술은 '계산표(計算表)'를 이용하는 것이었다. 소위 산술전문가에 의해서 계산된 결과들을 표로 만들어 보관하고 있다가, 필요할 때마다 자신이 원하는 계산결과를 찾아보는 것은 분명히 직접 계산하는 것보다는 편리했을 것이다. 독자적인 수체계를 사용하였으며 천문학적인 지식이 뛰어난 마야인들은 이미 BC 2000년경부터 특별히 고안된 곱셈표를 만들어 사용했던 것으로 알려져 있다. 흔히 '구구단'으로 불리우는 승법표(乘法表)는 곱셈을 필산함에 있어서 현재에도 널리 사용되는 하나의 좋은 예이다. 그러나, 계산해야 할 대상이 커짐에 따른 지속적인 표의 확장과 보관에는 많은 시간과 노력이 필요했으며, 사회에서 요구하는 계산내용도 점차 복잡해지면서 계산표를 만드는 작업 자체가 고도의 산술을 필요로 하게 되었다.

한편, 산판에 하나의 수를 표현하기 위해서는 각각의 자리수에 해당되는 칸마다 물건을 올려놓아야 했다. 처음에는 놓여진 물건의 개수가 나타내고자 하는 수와 서로 일치되었으나, 점차 숫자의 사용이 일반화되면서 산판에서 구슬이나 작은 돌멩이와 같은 물건들은 사라지고 숫자가 쓰여진 표식을 대신 사용하거나 아예 산판 위에 직접 숫자를 쓰게 되었다. 이렇게 진흙판이나 나무판과 같이 평평한 면(面)위에 모래같은 가루를 뿌리고 그위에 숫자를 쓰고 지우는 과정이 나중에 종이와 연필로 대체되면서 오늘날과 같은 필산으로 발전하게 되었다.

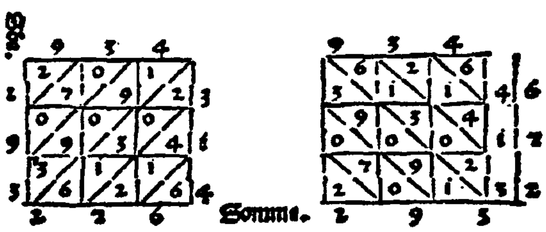
중세(中世) 유럽에서 널리 보급된 것으로 전해지는 제르베르식 산판은 기하적인 문제를 풀기 위하여 도형을 그리거나 계산을 하는 데에 주로 사용되었는데, 필산할 때에는 30개의 수직선을 그려 자리수를 구분하고 숫자가 새겨진 표식을 판위에 올려놓아 계산을 수행하였

다고 한다. 근대(近代)에 이르기까지 약 500여년동안 산판의 형태는 크게 바뀌었으며, 제르베르식 산판에 있었던 수직선과 숫자가 쓰여진 표식 대신에 왼쪽에서 오른쪽으로 수평선이 그어지고 주판의 구슬과 같이 아무런 숫자나 문자도 쓰여있지 않은 부자(符子)가 사용되었다.([2] pp.164-167) [그림 5]는 제르베르식 산판을 이용하여 4600×23이란 곱셈이 수행되는 과정을 나타낸 것으로, 숫자 '0'이 사용되지 않은 점을 제외하면 수의 위치는 조금 다르지만 현재 우리가 종이 위에서 행하는 필산과 별차이가 없음을 알 수 있다.



[그림 5] 제르베르식 산판을 이용한 곱셈계산([2] p.164)

위치적 기수법과 '0'(zero)을 사용했던 고대 인도인들이 산판을 이용하여 계산하는 방법은 매우 특이하다. 덧셈의 경우 제르베르식 산판과는 반대로 왼쪽부터 시작하여 오른쪽으로 이동해 가면서 계산을 진행하였으며([2] pp.164-167), 이러한 진행순서는 오늘날 주판을 이용한 덧셈계산과 일치하고 있다. 곱셈의 경우에는 '격자식 곱셈'(Shabacah)이라 불리는 더욱 독특한 형태의 계산방법을 사용하였는데, [그림 6]은 1478년 출간된 「트레비소 산수(算數)」에 나타난 격자식 곱셈의 한 예로써 934×314＝293,276이라는 계산결과를 보여주고 있다.

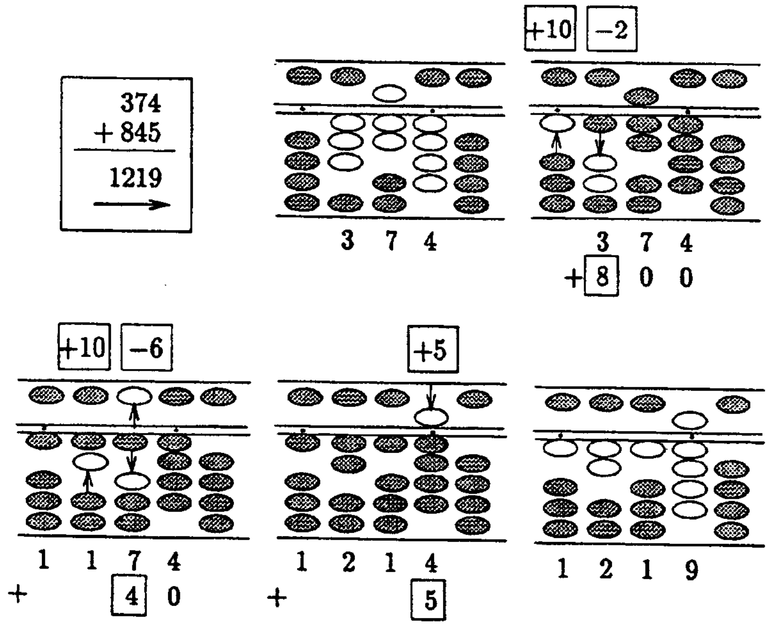


[그림 6] 고대 인도의 격자식 곱셈계산([2] p.208)

그림에서 보듯이 격자식 곱셈은 제르베르식 산판을 이용한 방법과는 달리 각각의 계산단계마다 숫자들을 지우는 불편을 없애 계산이 편리하고, 검산하기에도 용이한 장점을 가지고 있다. 또한, 각각의 격자를 채우는 작업이 서로 독립적이므로 순서에 상관없이 이루어질 수

있는 특징을 가지고 있다. 자리수간의 곱셈이 암송된 구구단에 의하여 이루어지는 것은 다른 계산방법과 같은 반면, 자리수간의 곱셈이 모두 처리된 뒤에 비로서 대각선 방향으로 덧셈이 행해진 것은 계산과정에서 덧셈과 자리수간의 곱셈이 반복적으로 나타나는 다른 계산방법과는 큰 차이가 있다는 사실에 주목할 필요가 있다.

주판을 이용한 계산방법에 대하여 살펴보기로 하자. 주판은 수를 표현하는 하나의 도구일 뿐만 아니라, 명수법에 의하여 불리워진 수를 주판에 나타내는 과정에서 덧셈이나 뺄셈에 대한 계산이 동시에 처리되는 특성을 가지고 있다. [그림 7]은 374+845=1,219라는 덧셈 계산이 진행되는 과정을 보여주고 있다. 우선 '삼백칠십사'라는 수를 나타내기 위하여, 들리는 순서대로 정해진 자리수에 '3', '7', '4'를 차례로 표시한다. 이제 845를 더하기 위하여, 먼저 '팔백'이라는 말에 따라 백(百)의 자리수에 '8'에 해당하는 주판알을 추가로 표시하도록 한다. 만일 주판알이 부족할 경우에는 천(千)의 자리수에 '1'을 더한 대신 백의 자리수에서 '2'를 빼도록 하여 '1000'과 800의 보수(補數)인 '-200'을 함께 표시함으로써 결과적으로 '팔백'을 더해주게 된다. 이러한 방식으로 '사십'과 '오'를 불려지는 순서대로 더한 뒤에, 주판에 최종적으로 표시된 수를 읽으면 '천이백십구'라는 덧셈계산 결과를 얻게 된다.



[그림 7] 주판을 이용한 덧셈계산

주판을 이용한 곱셈계산은 덧셈계산과 거의 유사한 방식으로 진행된다. 즉, 불려진 피승수(被乘數)와 승수(昇數)를 각각 주판에 표시한 뒤에, 구구단을 이용하여 가장 높은 자리수부터 내림차순으로 한 단계씩 자리수간의 곱을 차례로 표시하면서 덧셈계산을 수행하면 결과적으로 곱셈계산이 이루어지게 된다.

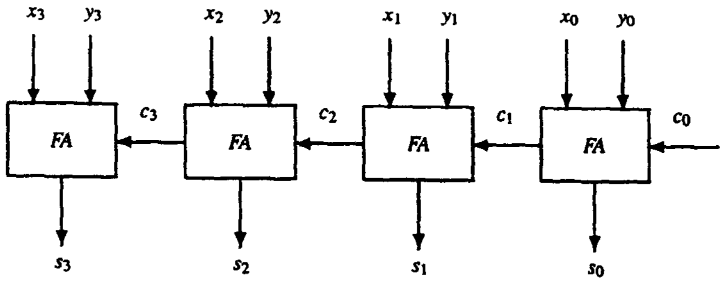
일반적으로 두 수의 합을 구하는 것은 거의 모든 산술연산에서 가장 자주 사용되는 기본적인 연산으로서, 덧셈과 뺄셈은 물론 곱셈과 나눗셈과 같은 복잡한 연산들의 계산과정을 단계별로 살펴보면 두 수의 합과 차를 구해야 하는 경우가 많다. 따라서, 산술의 발전과정에서 빠르고 정확한 덧셈계산을 위한 노력이 하나의 주축을 이루게 된 것이다. 이러한 사실은 컴퓨터의 경우에도 예외가 아니다. 다만, 필산이나 주판을 이용한 계산이 10진법에 근거를 두고있는 반면, 오늘날의 컴퓨터에서는 2진법을 사용한다는 것만 다를 뿐이다.

컴퓨터산술의 발전과정을 살펴보면 역사적으로 과거에 사람들이 사용했던 각종 수체계와 산술방식들이 모두 등장하고 있음에 놀라지 않을 수 없다. 컴퓨터에서의 덧셈계산이 지금까지 어떻게 발전되어 왔는 지 살펴보기로 하자. 임의의 정수, $X=(x_n x_{n-1}\cdots x_1 x_0)_2$와 $Y=(y_n y_{n-1}\cdots y_1 y_0)_2$가 주어져 있을 때, 두 수의 합, $S=X+Y=(s_n s_{n-1}\cdots s_0)_2$를 계산하는 것은 각각의 자리수간의 합, $x_i+y_i=(t_{i+1}t_i)_2$을 구하여 차례대로 모두 더하는 것과 같다. 이러한 자리수간의 합을 계산하는 데에는 FA(Full Adder)라는 논리연산회로(論理演算回路)가 사용되며,

$$s_i=x_i \oplus y_i \oplus c_i$$

$$c_{i+1}=(x_i \wedge y_i) \vee \{c_i \wedge (x_i \vee y_i)\}$$

라는 식에 따라 x_i와 y_i로부터 s_i와 c_{i+1}이 각각 생성된다(단, $c_0=0$). 이 때, c_{i+1}을 자리올림수(carry)라 부르며, 낮은 자리수의 합에서 생성된 c_i가 높은 자리수의 합을 구하는 데에 사용됨으로써 오른쪽에서 왼쪽 방향으로 자리올림수가 전파(propagation)된다. 이러한 방식의 덧셈계산을 실제 회로로 구현한 것이 [그림 9]에 나타낸 Ripple-Carry Adder이며, 자리올림수의 전파과정이 필산에 따른 2진법의 덧셈계산과 일치함을 보여주고 있다.

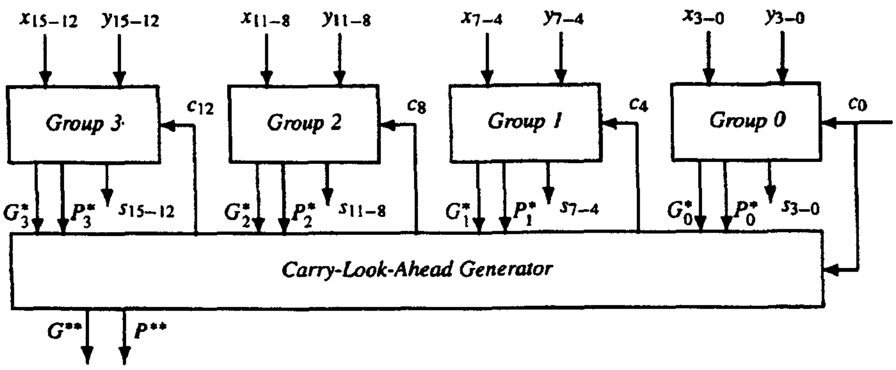


[그림 8] 4bit Ripple-Carry Adder

필산이나 주판을 이용한 계산에 있어서 발견되는 공통점 중의 하나는 계산의 대상(對象)인 수가 주어지는 방법에 있다. 즉, 사람들이 수를 부르거나 기록할 때 가장 높은 자리수부터 내림차순으로 진행하였으며, 이러한 순차적(sequential)인 수의 표현방식은 산술에서의

계산과정을 순차적으로 처리하게 된 가장 직접적인 이유가 되었다. 그러나, 천공카드방식에 따른 기계식 계산기나 진공관이나 트랜지스터를 이용한 전자식 계산기의 경우에는 계산과정에서 하나의 수를 인식함에 있어서 사람과는 달리 모든 자리수들을 동시에 병렬(parallel)로 받아들일 수 있게 됨으로써 사람에 의한 기존의 산술방식과는 근본적인 차이가 생겨났다. 따라서, 필산에 적합하지 않았던 수체계나 산판 또는 주판을 이용한 계산방법들도 컴퓨터산술의 측면에서는 충분히 고려할만한 가치가 있게 된 것이다.

앞의 Ripple-Carry Adder에서 보듯이 비록 모든 자리수, x_i와 y_i가 동시에 주어짐에도 불구하고 자리올림수의 전파는 컴퓨터에서의 빠른 덧셈계산을 막는 가장 큰 장애물이었다. 사실 지난 수십년동안 이루어진 컴퓨터산술의 발전과정은 자리올림수와의 전쟁이라고 해도 과언이 아니다. 일반적으로, 자리올림수의 전파는 '1+1'의 계산자체에서 발생하는 경우와 '0+1' 또는 '1+0'일 때 자체적으로는 자리올림수가 발생하지 않으나 전단계에서 발생한 자리올림수가 전파된 경우의 두 가지로 나누어 볼 수 있다. 오직 '0+0'인 경우에만 자리올림수가 발생하지도 전파되지도 않는다. 따라서, 동시에 병렬로 주어진 모든 x_i와 y_i를 가지고 자리올림수의 발생과 전파를 미리 예상하여 자리올림수의 전파에 따른 지연시간을 줄이려는 연구가 진행되었다. [그림 10]은 16bit Carry-Look-Ahead Adder(CLA)로서, 그림에서 x_{3-0}은 x_3, x_2, x_1, x_0의 4bit의 그룹을 나타내고, G_i^*와 P_i^*는 각각 4bit의 그룹별로 자리올림수의 발생과 전파를 그리고 G^{**}, P^{**}는 각각 전체 16bit의 자리올림수의 발생과 전파를 나타낸다.



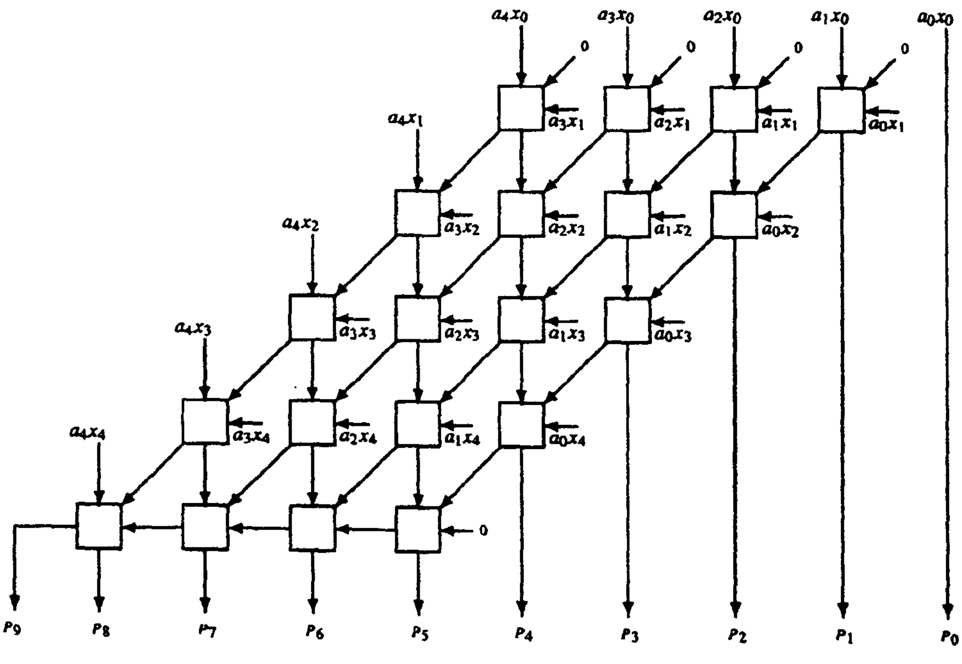
[그림 10] 16bit Carry-Look-ahead Adder

이러한 방식의 자리올림수 처리방법은 주판을 이용한 덧셈계산에서 미리 자리올림수를 표시한 뒤에 보수를 계산하는 것과 같은 이치(理致)이다. CLA이후의 새로운 덧셈기의 개발은 VLSI기술의 발전과 밀접한 관계가 있으나, 산술의 관점에서는 모두 자리올림수의 전파에 따른 지연시간을 최소화기 위한 다각적인 노력의 결과이다. 예를 들면, 여러 개의 인접한 자

리수에서 $x_i \neq y_i$이라는 조건이 만족되면 자리올림수의 전파과정이 일부 생략될 수 있다는 점을 이용한 Carry-Skip-Adder나 부분합을 구하는 과정과 자리올림수의 전파과정을 분리시켜 동시에 처리하도록 한 Carry-Save Adder 등을 들 수 있다. 이와 같이, 컴퓨터에서의 덧셈계산은 모든 자리수가 동시에 주어짐에 따라 계산과정에서 나타나는 자리올림수의 전파과정이 합(合)을 구하는 과정과 동시에 병렬로 처리된다는 특징을 가지고 있다.

한편, 컴퓨터에서의 곱셈계산과 고대 인도에서 사용되었던 격자식 곱셈을 비교해보면 그 당시 인도사람들의 완벽한 계산방식에 감탄하지 않을 수 없다. 사실 격자식 계산방법은 직사각형의 격자형태를 평행사변형으로 약간 변형해보면 현재 우리가 손으로 2진법에 의한 곱셈을 계산하는 것과 마찬가지 임을 알 수 있으며, 이것은 또한 [그림 11]에서 보듯이 컴퓨터에서 배열(配列, array)을 이용한 곱셈기(multiplier)의 계산방식과도 일치하고 있다.

					a_4	a_3	a_2	a_1	a_0
				×	x_4	x_3	x_2	x_1	x_0
					$a_4 \cdot x_0$	$a_3 \cdot x_0$	$a_2 \cdot x_0$	$a_1 \cdot x_0$	$a_0 \cdot x_0$
				$a_4 \cdot x_1$	$a_3 \cdot x_1$	$a_2 \cdot x_1$	$a_1 \cdot x_1$	$a_0 \cdot x_1$	
			$a_4 \cdot x_2$	$a_3 \cdot x_2$	$a_2 \cdot x_2$	$a_1 \cdot x_2$	$a_0 \cdot x_2$		
		$a_4 \cdot x_3$	$a_3 \cdot x_3$	$a_2 \cdot x_3$	$a_1 \cdot x_3$	$a_0 \cdot x_3$			
	$a_4 \cdot x_4$	$a_3 \cdot x_4$	$a_2 \cdot x_4$	$a_1 \cdot x_4$	$a_0 \cdot x_4$				
p_9	p_8	p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

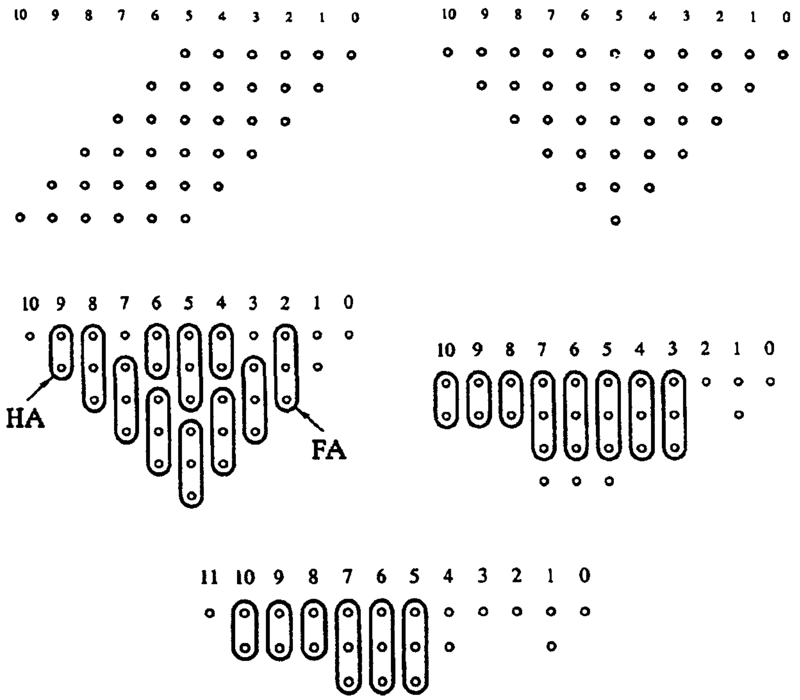


[그림 11] 필산과 array multiplier에서의 곱셈계산방식의 비교([6] pp.115-116)

이 때, 2진법에서의 자리수간의 곱은 '0'일 때에는 무시하고, '1'일 때에만 '1'의 자리수만큼 왼쪽으로 이동(移動, shift)하는 것으로 쉽게 얻어질 수 있다. 같은 방법으로 두 정수의 곱을 계산하는 것은

$$X \times Y = X \times \sum_{i=0}^{n} y_i 2^i = \sum_{i=0}^{n} (y_i \times X) 2^i$$

와 같이 여러 개의 부분곱(partial product), 즉 $y_i \times X \times 2^i$들의 합을 구하는 것과 같다. 따라서, 컴퓨터에서의 곱셈계산은 덧셈계산과 마찬가지로 부분곱들의 덧셈계산을 효율적으로 처리할 수 있도록 발전해 왔다. 일례로 [그림 12]는 여섯 개의 부분곱들의 합이 각각의 자리수에 따라 병렬로 계산되는 과정을 나타내고 있다. 그림에서 기호 'o'은 '0'또는 '1'과 같은 하나의 이진숫자(binary digit)를 나타내며, 맨위의 0부터 10까지의 숫자 k는 자리수 2^k를 의미한다. 이진숫자들이 2개 혹은 3개씩 묶여있는 것은 각각 HA(Half Adder)와 FA(Full Adder)라는 논리연산회로에 의한 덧셈계산을 나타내며, HA는 FA와 달리 두 개의 이진숫자들의 합, $x_i + y_i = (c_{i+1} s_i)_2$을 계산하는 논리연산회로로서, $s_i = x_i \oplus y_i$와 $c_{i+1} = x_i \wedge y_i$라는 식에 따라 x_i와 y_i로부터 s_i와 c_{i+1}이 각각 생성된다.



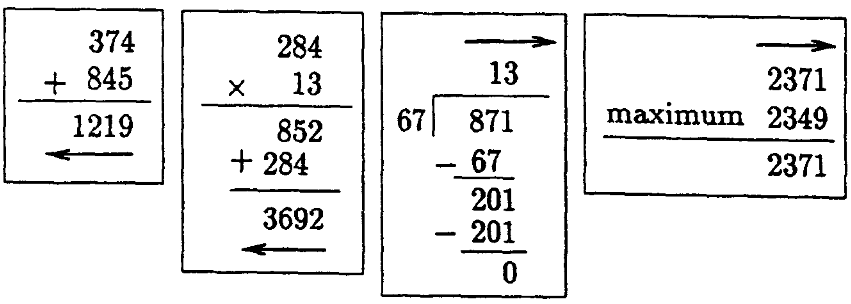
[그림 12] HA와 FA를 이용한 곱셈계산의 병렬처리

이밖에도, 여기에서는 자세히 설명하지 않겠지만, 컴퓨터에서의 나눗셈이나 삼각함수의 계산 등과 같이 더욱 복잡하고 다양한 산술연산의 처리과정은 과거에 사람들이 필산이나 각종 산술도구를 이용한 계산을 통하여 보여 준 많은 아이디어들이 반도체 기술과 어우러져 새로운 모습으로 활용되고 있음을 보여주고 있다. 일례로 '계산표'를 하나의 산술도구로 볼 때, 컴퓨터에서는 삼각함수나 지수함수, 로그함수 등의 함수값을 계산하는 데에 계산표를 사용함에 따른 장점을 살려 'Table-lookup' 방식이 사용되고 있다.([8] pp.110-116)

그러나, 컴퓨터산술의 발전과정은 기존의 산술과는 달리 사람들이 필산이나 주판과 같은 산술도구를 이용한 계산을 간편하고 쉽게 할 수 있도록 하기 위한 것이 아니라, VLSI 설계이론과 제조기술에 따른 회로설계상의 효율성과 실제 구현(implementation)을 위한 실용성에 따라 크게 좌우되어 왔다는 사실을 간과해서는 안된다. 즉, 인간중심의 산술이 컴퓨터의 등장에 따라 기계중심의 산술로 변화되어 발전된 것이 소위 '컴퓨터산술'이라는 새로운 산술의 역사를 만들어가고 있는 것이다.

Ⅳ. 산술(算術)의 미래(未來)

위치적 기수법에 따른 산술은 각각의 연산의 특성에 맞도록 항상 일정한 방향성을 가지고 순차적으로 계산되어 진다. 필산의 경우, [그림 13]에서 보듯이 덧셈과 곱셈은 덧셈계산에서 발생하는 자리올림수의 전파로 인하여 가장 낮은 자리수부터 높은 자리수 방향으로, 나눗셈과 Maximum과 같은 수의 비교는 가장 높은 자리수부터 낮은 자리수 방향으로 계산이 진행된다.



[그림 13] 다양한 산술연산의 필산과정과 계산방향

그러나, 주판을 이용한 산술은 고대 인도인의 계산방법과 마찬가지로 덧셈이나 곱셈까지도 가장 높은 자리수부터 낮은 자리수 방향으로 계산이 진행된다. 물론, 필산의 경우나 주판을 사용하는 경우 모두 위치적 기수법인 10진법을 사용하고 있다. 그런데, 계산방향은 어떻게 서로 다를 수 있는가? 또한, 무슨 목적 때문에 서로 다르게 되었는가? 이러한 의문을 풀기 위하여 주판을 이용한 덧셈계산을 다시 한번 음미해 볼 필요가 있다.

주판을 이용한 덧셈계산에서는 자리수마다 표현될 수 있는 구슬의 개수에 따라 선택적으로 보수(補數)를 사용하고 있다. 이러한 방법으로 자리올림수의 발생 자체를 방지하여 덧셈계산을 가장 높은 자리수부터 차례로 처리할 수 있었던 것이다. 앞장의 예에서 보았듯이 374에 845를 더하기 위하여 실제로는 1105를 더함과 동시에 260을 뺀 것이다. 뺄셈기호인 '-'와의 혼동을 피하면서 하나의 자리수로서의 음수(陰數)를 표시하기 위하여 숫자 위에 '-'(bar)를 붙여 $\bar{1}=-1$, $\bar{2}=-2$, … 와 같이 나타내기로 하면,

$$1\bar{2}00+1\bar{6}0+5=1000-200+100-60+5=845$$

라는 사실을 확인할 수 있다. 그렇다면, 주판을 사용함에 있어서 이와 같이 복잡한 보수를 사용하면서까지 계산방향에 집착한 이유는 무엇일까? 그것은 더해질 수가 불리어지는 순서대로 계산하기 위해서였을 것이다. 만약 '일억이천삼백사십오만육천칠백팔십구(123,456,789)'와 같이 길이가 긴 수를 듣고난 뒤에 가장 낮은 자릿수부터 계산해야한다면 이미 들은 수를 주판위에 별도로 표시하거나 모두 머리속에 암기해 두어야야만 한다. 이 또한 계산 못지않게 얼마나 번거럽고 귀찮은 일인가? 더구나 더해질 수의 마지막 자리수를 들을 때까지 덧셈계산은 시작조차 할 수 없게 된다. 따라서, 명수법에 맞도록 가장 높은 자리수부터 차례로 덧셈을 계산하는 것이 필요했던 것이다.

주판 위에서는 수를 더하거나 빼는 작업이 공통적으로 구슬위치의 상하(上下)변동에 의하여 이루어지므로, 두 손가락-엄지와 검지를 사용하여 보수를 이용한 덧셈계산과 같이 동시에 두 개의 서로 다른 자리수의 계산을 처리할 수 있다. 물론, 7이나 8처럼 5보다 큰 수를 표현하기 위하여 윗칸에 있는 5를 의미하는 구슬과 아랫칸에 있는 1을 의미하는 구슬들을 한꺼번에 움직이는 데에도 엄지와 검지는 유용하게 사용되었다. 이와 같은 주판사용을 위한 운지법(運指法)은 궁극적으로 계산과정에 있어서 손과 손가락의 움직임을 최소한으로 줄여줌으로써 빠르고도 정확한 계산이 이루어질 수 있도록 고안된 것이다.

음수(陰數)를 이용한 위치적 기수법은 이미 1000년 전부터 인도에서 사용되었던 것으로 알려져 있다. 근대(近代)에 들어서면서, 코시(A. Cauchy)는 10진법에서 0부터 9까지의 숫자 대신에 -5부터 5까지의 숫자를 이용하여 수를 표현하면 '구구(九九)단' 대신에 '오오(五五)단'만으로도 곱셈계산이 가능하다는 점을 지적했으며, 라레인(L. Lalane)은 소위 '대칭적 3진법'(balanced ternary notation)을 이용하여 0, 1, 2대신에 -1, 0, 1을 가지고 모든 수를 표현할 수 있는 새로운 수체계를 제시하였다. 1960년 러시아에서 실험적으로 제작된 '세턴(SETUN)'이라는 컴퓨터는 이러한 대칭적 3진법을 채택한 최초의 전자식 계산기였으며, 크누스(D.E. Knuth)는 그의 저서에서 미래에 전자를 대신하여 'flip-flop'대신에 'flip-flap-flop'을 표현할 수 있는 새로운 매체가 등장하게 되면 대칭적 3진법은 매우 유용한 수체계가 될 수도 있을 것이라고 예상하였다.([5] pp.190-192)

1961년 아비제니스(A. Avizienis)는 음수를 이용한 위치적 기수법을 모든 진수에 대하여 일반화(一般化)하여, 하나의 수체계를 구성하기 위하여 진수와 숫자들의 집합간에 어떠한 관계가 있는지를 밝혀주었다. 또한, 그는 이러한 수체계를 사용하면 덧셈이나 곱셈에서 자리올림수의 전파없이 계산이 가능하다는 점을 지적함과 동시에, 각각의 자리수에 대한 합산과정을 병렬로 처리할 수 있는 알고리즘을 발표하였다.([1])

일반적으로 r진법에서 0부터 $r-1$까지의 숫자 대신에 $\overline{r-1},\cdots,\bar{1},0,1,\cdots,r$을 사용

하면 비록 같은 수라 할지라도 두 가지 이상의 서로 다른 표현이 가능하게 된다. 예를 들어, $r=10$인 경우에

$$845=1\,\bar{1}\,\bar{6}\,5=1\,\bar{2}\,4\,5=9\,\bar{6}\,5=8\,5\,\bar{5}$$

가 성립한다. 이와 같이 보수를 이용한 표현의 중복(重複, redundancy)가능성은 주판을 이용한 덧셈계산에서 보았듯이 자리올림수의 발생을 방지하는 데에 효과적으로 이용될 수 있다. 그러나, 지나치게 중복을 허용하는 것은 오히려 수의 표현 자체 뿐만 아니라 산술연산을 위한 계산상의 복잡도를 증가시키는 요인이 되므로, 수의 표현에 사용되는 숫자들을 적당히 제한할 필요가 있다.([1])

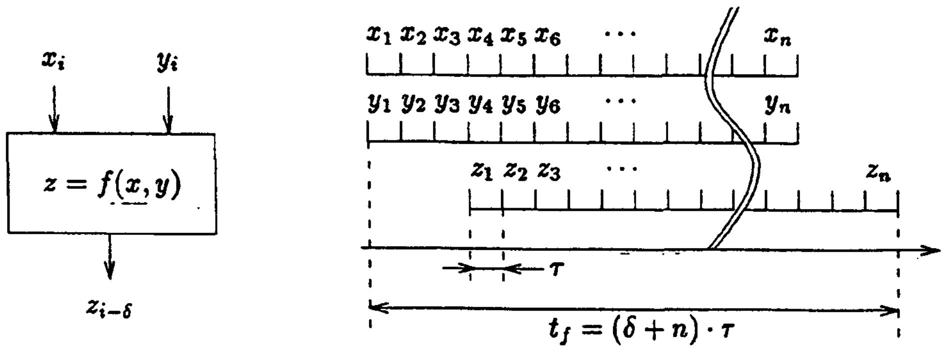
특히, $r=2$이면 $\bar{1}(=-1)$, 0, 1을 사용하는 것이 유일한 선택이며, 앞에서 말한 대칭적 3진법과는 혼동하지 않도록 구분할 필요가 있다. [그림 14]는 이와 같이 변형된 2진법(signed binary number representation)을 사용함으로써, 30+29=59라는 덧셈계산이 자리올림수의 전파없이 어떻게 실행될 수 있는 지를 보여주고 있다.

i	0	1	2	3	4	5	6	7	Number
x_i		1	0	$\boxed{0}$	$\boxed{\bar{1}}$	1	0	(0)	30
y_i		1	0	$\boxed{\bar{1}}$	$\boxed{1}$	0	1	(0)	29
t_i	1	0	$\boxed{0}$	0	1	0	(0)		68
w_i	(0)	0	0	$\boxed{\bar{1}}$	0	$\bar{1}$	1		−9
s_i	1	0	0	$\bar{1}$	1	$\bar{1}$	1		59 (= 30 + 29 = 68 − 9)

[그림 14] 음수를 이용한 변형된 2진법에서의 덧셈계산

이러한 새로운 수체계는 비단 자리올림수의 전파없이 덧셈계산이 가능하다는 점 이외에도 컴퓨터산술 분야에서 주목할 만한 여러 가지 특성들을 가지고 있다. 우선 두 수의 합을 계산함에 있어서, 각각의 자리수들이 높은 자리수부터 하나씩 차례로 주어지면 계산결과도 높은 자리수부터 하나씩 얻어지는 순차적인(digit-serial) 덧셈계산이 가능하다. 이러한 계산방식은 덧셈 이외에도 곱셈이나 나눗셈, 지수·로그함수나 삼각함수까지도 모두 적용될 수 있으며, 이를 소위 '온라인산술(on-line arithmetic)'이라 부른다.([8],[9]) 이에 따른 새로운 산술방식은 컴퓨터를 이용한 수치계산(numerical computation)의 정확도(precision)를 하드웨어의 확장없이 획기적으로 개선할 수 있다. 또한, 대수식(代數式, algebraic expression)과 같이 여러 개의 서로 다른 산술연산들이 이어질 경우 자리수 단위(digit-level)로 파이프라인(pipeline)방식에 따라 각 단계별 계산과정들이 동시에 병렬로 처리됨으로써 전체 대수식의

계산시간을 대폭 줄일 수 있다.([3],[4]) [그림 15]는 온라인산술의 계산방식을 나타낸 것으로서, 한 자리수간의 계산주기(period)인 τ와 계산에 사용된 자리수와 계산결과 얻어진 자리수간의 차이(delay)인 δ에 의하여 전체 산술연산의 계산시간이 결정된다.



[그림 15] 온라인산술의 계산방식과 특성

현재는 0과 1이외에 $\bar{1}$을 표현하기 위하여 하나의 숫자에 2 bit가 필요하지만 만일 전자 이외에 빛이나 레이저와 같은 새로운 매체를 통하여 표현방법이 물리적으로 단순해진다면 이에 따른 산술연산은 초병렬(massively parallel)인 동시에 대규모 파이프라인(pipeline)이 가능한 새로운 고성능(high performance) 컴퓨터 구조에 응용될 수 있다. 지금까지의 병렬 컴퓨터에 대한 연구는 주로 다중(多衆) 프로세서들간의 내부연결 위주로 이루어진 데에 비하여, 미래에는 하나의 프로세서 내에 종류도 서로 다르고 개수도 엄청나게 많은 산술연산자(arithmetic operator)들이 다이나믹하게 서로 연결되는 이른바 사람의 신경망(神經網)과 같은 컴퓨터가 등장하게 될 것이다.

V. 결 론

최근에 우리 주변에서 가장 널리 보급된 문명의 이기(利器)들 중에 대표적인 것으로 '자동차'와 '컴퓨터'를 들 수 있다. 이러한 발명품들을 사용할 줄 아는 사람들은 점점 많아지지만, 그 내부에 들어있는 각각의 구성요소나 동작원리에 대하여 자세히 알고 있는 사람들은 극히 드물다는 면에서 자동차와 컴퓨터는 묘한 공통점을 가지고 있다.

우리는 역사를 통하여 문명(文明)은 생각(idea)과 기술(technology)의 만남에 의해서 발전되어 왔음을 확인하고 있다. '사람과 물건을 편안하게 이동시켜줄 수 있는 기계'에 대한 아이디어와 엔진 개발기술이 결합하여 오늘날 자동차가 널리 사용되고 있는 것처럼, '사람 대신 계산을 빠르고 정확하게 처리할 수 있는 기계'를 만들기 위한 줄기찬 노력은 반도체 개

발기술과 접목되어 단순히 산술도구라기보다는 이제는 디지털문명의 필수품이 되어버린 컴퓨터가 급속도로 보급되고 있다. 이러한 의미에서, 과연 '산술(算術)'이란 무엇이고 어떻게 발전해 왔는지 살펴보는 것은 컴퓨터의 발전과정을 이해하기 위한 가장 손쉬운 방법 중의 하나일 것이다.

필산의 용이함으로 인하여 수천년에 걸쳐 사람들에게 선택되고 발전된 10진법, 그리고 그에 따른 산술이 수학이라는 학문의 출발이라면 천공카드와 진공관에 의해 도입된 2진법, 그리고 그에 따른 산술은 컴퓨터의 구조와 동작원리를 이해하고 나아가 전자계산학(computer science)의 발전과정을 살펴보기 위한 첫걸음이 될 것이다. 인간중심의 산술에서 기계중심의 산술로 바뀌어버린 지금, 새로운 수체계와 그에 따른 산술에 대한 연구는 단순히 산술 자체가 아니라 인간의 삶에 미칠 엄청난 영향과 파급효과를 고려하면 일종의 미래학(未來學)이라는 생각마저 갖게된다.

주판이 점점 자취를 감추는 현실과는 대조적으로 주판을 이용한 산술에 깔려있는 계산철학은 산술의 발전과정이 컴퓨터산술이라는 새로운 분야에서 다시 재현되고 있음을 보여주는 일종의 연결고리인 것이다. 흔히, 학교교육에서의 컴퓨터교육은 PC를 이용하여 DOS나 Windows같은 운영체제나 워드프로세서같은 소프트웨어 사용법을 학습하는 것으로 인식하고 있으나, 사실 가장 중요한 것은 디지털시대가 열리게 된 배경과 컴퓨터 본래의 목적인 산술연산의 계산능력을 이해하는 것이 더욱 중요하다고 볼 수 있다. 마치 대부분의 사람들이 유치원부터 대학을 졸업할 때까지 수십년간 입시(入試)에 왜곡된 수학을 공부한 결과 수학의 본질이라 할 수 있는 논리적이고 추상적인 사고(思考)를 제대로 할 수 없는 것처럼, 원리나 이유를 모르는 채 반복적인 훈련만으로 일관된 컴퓨터교육은 인간을 더욱 기계적으로 만들 수 있기 때문이다.

참고문헌

1. A. Avizienis, *"Signed-digit number representations for fast parallel arithmentic"*, IRE Trans. Eletron. Comput, Vol. Ec-10, no.3, 1961, pp. 389-400.
2. F. Cajori, A History of Elementary Mathematics, 정지호 , 캐조리 數學史, 창원사, 1977.
3. M. D. Ercegovac, *"On-line Arithmetic: A design methodology and Applications"*, IEEE Trans. comput, vol.C-26, no.7, 1988, pp.667-680.
4. M.J. Irwin and R.M. Owens, *"Digit-pipelined arithmetic as illustrated by the paste-up system: A tutorial"*, Computer, avnl, 1987, pp.61-73.
5. D.E. Knuth, The Art of Computer Programming, 2nd Ed, Vol 2 : Seminumerical Algorithms, Addison-Wesley Pub. Co, 1981.
6. I. Koren, Computer Arithmetic Algorithms, Prentice-Hall, Inc, 1993.
7. 이상범, 이규영, 전산학 개론, 개정판, 정익사, 1996.
8. J.-M. Muller, Computer Arithmetic : Operators and Elementary Functions, Masson,

1989, *in French.*
9. H.-J. Yeh, Parallel Algorithmes and Architectures based on Digit-Serial Arithmetic Computation, Ph.D. Thesis, Ecole Normale Supérieure de Lyon, 1993, *in French.*

수학사와, 수리철학, 그리고 현재의 문화에서 수학의 역할에 대한 지식과 경험을 선생님들이 가지고 있지 않다면, 어떻게 선생님들이 교육과정을 짜고, 교과서를 선택하고 동료나 학부형에게 수학을 설명하거나 학생들에게 수학에 대한 제대로 된 그림을 보여주리라고 기대하겠는가?

— K. 메이

생물학자, 통계학자, 수학자, 컴퓨터학자가 아프리카에 사진 탐험중이었다. 지프차를 타고 대초원으로 나가 쌍안경으로 지평선을 살펴보고 있었다.

생물학자 : "보시오! 얼룩말 한 떼가 있군요
그리고 거기 한가운데에 하얀 얼룩말이 한 마리 있어요
대단한데요! 하얀 얼룩말이 있다니! 우린 유명해질거요!"
통계학자 : "별로 중요한 일이 아니요. 우린 단지 하얀 얼룩말 한 마리가 있다는 걸 아는 것 뿐이요."
수 학 자 : "실제로 우리는 한쪽 면이 하얀 얼룩말이 한 마리 존재한다는 것을 알 뿐이요."
컴퓨터학자 : "이런 예외적인 경우군!"