PERGAMON Computers and Mathematics with Applications 39 (2000) 89–93

An International Journal
computers & mathematics
with applications

www.elsevier.nl/locate/camwa

A Brief History of the Development of Error Correcting Codes

I. S. REED
Communication Sciences Institute
University of Southern California
Los Angeles, CA 90089-2565, U.S.A.

Abstract—This paper is based on a talk the author gave at the symposium in honor of Solomon Golomb's 60th birthday. It describes the history in which the author was a participant in some of the early developments of error-correcting codes and computers. Also details are given about the origins of both the Reed-Muller and Reed-Solomon codes.

I don't know where to begin. I suppose that the history of anything on coding would have to start with something I know best and that would be me. I guess I could start back at Caltech, but I won't for now. Actually, the first people that I knew in coding other than myself were David Muller and David Huffman. David Muller and I were classmates at Caltech. David Huffman and I met while on the same ship in the U.S. Navy. During the early 1940s none of us knew anything about error-correcting codes; they had not yet been discovered.

David Huffman was my officer and I was his petty officer aboard a U.S. Navy destroyer. Except possibly for past President George Bush, Ensign Huffman well may have been the youngest officer in the U.S. Navy although President Bush was quite young as well. I was trained to be a U.S. Navy radio technician as also, somewhat later, was coding expert Professor Lloyd Welch also at USC. Lloyd and I went to the same Navy technician school for radio and radar. That is how we both came to understand electronics.

My first knowledge of error-correcting codes was after finishing graduate school at Caltech in mathematics. I had gone to work for the Northrop Aircraft Corporation. As a graduating Ph.D. student in mathematics I had received offers to teach mathematics at such universities as Washington University in St. Louis, the University of Minnesota, etc. The former would not have been too bad. I had grown up in Fairbanks, Alaska, and Minnesota just seemed too cold for me to go there. However, I decided to stay in Los Angeles because Northrop was the organization that paid me the most money. Economics became the deciding factor in my decision.

At that time Northrop Aircraft had an excellent procedure for obtaining technical journals. They would either buy or borrow journals from libraries such as UCLA, Caltech, etc. I believe their standard place to obtain circulating journal articles was from the UCLA Library. Journals would come across my desk quite often.

One might imagine that in those days engineers worked in nice offices with walls, windows, etc. Actually, we worked in a very large area in a "sea" of drafting tables and desks. The noise level was at least 20 to $30dB$ higher than anything a person would work in today.

0898-1221/00/$ - see front matter Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX
PII: S0898-1221(00)00112-7

One journal that came to my desk then was the recent IRE Proceedings Journal of June, 1949. In this journal the first published paper on error-correcting codes came to my attention. It was the famous Letter-To-The-Editor by Marcel Golay. Later someone showed me a then somewhat obscure paper by Shannon, his 1949 paper in the Bell System's Technical Journal. These articles did not make too much of an impression on me since at the time I was working on how one might guide a cruise missile, called the SNARK missile.

I helped first on the guidance system of SNARK which was ultimately built. However, it never worked too well. Missiles fired much later from Cape Canaveral sometimes went off course to such remote areas as the rain forests of Brazil. This happened so often that they were called the SNARK infested jungles of Brazil. I never actively worked on control theory again.

I also contributed to an early digital computer suggested for use to control SNARK. It was called MADDIDA for magnetic drum digital differential analyzer. It was developed in 1949 by the Northrop guidance group in which I worked. It is my understanding that MADDIDA was the first electronic digital computer ever built on the West Coast.

There is an interesting story about the name MADDIDA. To let the world know of the existence of this machine, the computer was demonstrated at the first ACM meeting in 1950. Some people at this conference said that the Northrop group had named it Mad Ida for the somewhat derogatory name given to the brilliant and excitable lady programmer, Dr. Ida Rhodes. She was in charge of the software of SWAC, the first computer at the National Bureau of Standards. Actually, none of us from Northrop on the west coast had known before of this remarkable lady.

My next association with codes other than computer address codes occurred at the MIT Lincoln Laboratory when I became acquainted with Mr. Oliver Selfridge. Oliver Selfridge worked in Bill Davenport's group at Lincoln Laboratory. Originally, Lincoln Laboratory was formed at MIT to study the Air Defense system of the U.S. It was established by Professor George Valley of MIT in 1950.

Oliver Selfridge's job was to look for code sequences which could be transmitted on a communications system, called NOMAC. This system was supposed to have both a self-synchronization property and a modulation which looked like random noise. Today such a modulation scheme is called a spread spectrum waveform. At that time my work was with digital computer design. I had found previously an equation which describes mathematically the nature of the state diagram of a single two-state digital device. The time-difference equation associated with this bi-stable device was called by me the flip-flop equation.

My mathematical study of digital devices occurred somewhat before Oliver Selfridge told me of Sol Golomb's early work at the Glen L. Martin Corporation on linear sequence generators. The flip-flop equation was equivalent to the generator equation of a first-order polynomial sequence. In fact, the flip-flop can be considered to be a linear polynomial operator of unit delay, which operates on a digital time sequence.

Although I did not meet Sol Golomb at that time, his work made me realize what a good idea I had to describe digital logic in terms of time-difference equations. This reminds me of an anecdote I have to tell you about Sol Golomb.

About 22 years ago I was invited to go to Israel to give lectures on Reed-Solomon codes. I gave these talks at a company in Tel Aviv, called the Tadaran Corporation. They kept me so busy that I had little time to visit anyone. Sol had given me the telephone number of his older sister. Since I didn't have time to see her, I talked to her on the telephone. She sounded exactly like Sol except that her voice had a somewhat higher pitch.

I asked her, "You undoubtedly must be quite good at mathematics like your brother."

She said, "No. My field is English. I have only a limited mathematical ability compared with Sol." She told me that Dr. Golomb as a boy was a mathematical prodigy. As a young man he advanced so rapidly in mathematics that few of his teachers knew enough to teach him.

Let me back-track somewhat to Shannon. Shannon at MIT had worked for Professor Caldwell when he was a graduate student. He had come from upstate Michigan to MIT and while in

graduate school got the job of running the differential analyzer (DA) invented by Vannevar Bush. The differential analyser was a very large mechanical analog computer which covered the considerable area of several rooms.

The Vannevar Bush DA consisted of twenty wheel-and-disc integrators, many gears, shafts, etc. Shannon got his first knowledge of computation and information theory from his work on this machine. Shannon's early work on digital logic resulted partly from his work on the large number of relay circuits that were in the DA.

During WW2, for example, the start-circuit of a radar was actually a small relay computer. In fact, all early digital machines including pin-ball machines, were relay circuits, even the large telephone exchange circuits. Such circuits sometimes failed, and we as radar technicians were trained to deduce the cause. Both David Huffman and I had learned this from our Navy experience. The Navy training helped Huffman as a graduate student at MIT under the guidance of Professor Caldwell to develop a new technique for systematically designing relay circuits. Huffman's design method is still used today in the design of VLSI circuits. He showed the world of engineering how to create relay circuits with a minimum number of states, a very clever algorithm, indeed. Later as both a student and professor at MIT, he also contributed to the subject of linear sequential circuits and invented the Huffman codes which are now so important to image data compression in what is called the JPEG algorithm.

My work on computer logic at MIT Lincoln Laboratory led me in 1952 and 1953 to deliver an in-house lecture course on computer logic and design. After giving the lab an introduction to computer logic and what is now called the register-transfer languate (RTL) method of computer design, I developed for these lectures some notes on the elements of coding theory. These lecture notes were based on Hamming's famous 1950 paper in the BSTJ.

Shortly after I gave these lectures David Muller sent me a copy of his University of Illinois report he had written about computer logic and its applications to coding. David and I attended many of the same classes at Caltech and early on our U.S. Navy careers overlapped. He was trained as a theoretical physicist, but he really excelled at mathematics. Maybe it was his genetics: his mother was a mathematician, and his father was Herman Muller who got the Nobel Prize in genetics for discovering the gene. After a thesis on experimental physics David Muller became involved in logic design and its applications to digital systems.

To do this David Muller went to the University of Illinois to what was called the control systems laboratory. This laboratory was run by Professor Loomis, the former assistant director of the radiation laboratory at MIT during WW2. Before Dave sent me his report, I gave him a copy of my notes on computer logic. From the receipt of my notes on computer logic he became very interested in these computer design methods. Previously, he had invented his own version of computer logic. His early work was quite original and did not use the standard notation and techniques of logic and modern algebra. I, of course, at Caltech had had an excellent course on mathematical logic given by Professor E.T. Bell as well as an outstanding course on modern algebra given by Professor R.P. Dilworth.

In a notation of his own invention, Dave Muller described a new error-correction code in his report on logic design. His codes were based on what he called a Boolean Net Function which I didn't have the energy at the time to understand. However, I decided what he must have had in mind were what are called multinomials over a Boolean field, the finite or Galois field $GF(2)$ of two elements 0 and 1. From Professor Robert Dilworth at Caltech, I had learned much about algebraic ring theory and polynomial over fields. Professor Dilworth was one of the principal experts in modern algebra at the time. At one point I considered doing my dissertation under him, but decided that it might take too long, particularly since I only had enough money left on the G.I. bill to just complete my education in mathematics.

My idea of using polynomials over the primitive finite field $GF(2)$ of two elements was very fruitful. By constraining the maximum degree of these multinomials over N variables I managed to construct an error-correcting code which was equivalent to the codes Muller had found. The

algebraic structure imposed on these multinomials made it possible for me to find a decoding algorithm for these codes which is now called majority-logic decoding. Also, I demonstrated that these codes are group codes. They are a group or vector space with respect to vector addition over $GF(2)$.

Originally, the paper on these codes, now called Reed-Muller codes, was published as the MIT Lincoln Laboratory Report, No. 44, titled, "A Class of Multiple Error Correcting Codes and the Decoding Scheme," in mid-1953. At the time, I knew of no appropriate journals in which to publish new results in the field of coding. As a consequence the report lay dormant for almost a year. But one day I got an excited call from Professor Bob Fano in the school of engineering at MIT. He wanted me to come down to the Institute, more than ten miles away, the very next day and present this report at his weekly seminar on information theory. The next day I got a talk together and presented to Bob Fano and his graduate students the first seminar on the Reed-Muller codes. Dr. Fano was so enthusiastic about this seminar that he made sure that the same material was presented at the first Symposium on Information Theory, Cambridge, MA held September, 1954.

At the symposium I met Marcel Golay and briefly talked with Claude Shannon, the discoverer of information theory. Of course, Marcel Golay had discovered the first example of a multiple error-correcting code, namely the perfect (23,12) Golay code which corrects three errors. Hamming had discovered the entire class of one error-correcting codes. As a consequence when Marcel Golay told me that he was very impressed by my paper, titled "A class of multiple-error-correcting codes and the decoding scheme," I felt quite proud. It was never again my privilege to meet Marcel Golay. However, I had occasion to use both his radar codes and his very important three-error correcting code.

My next interaction with a future coding theorist was Dr. Neal Zierler, who recently retired from IDA, Princeton. Remarkably Dr. Zierler is given the credit for the invention of the first FORTRAN-like language for a computer. In 1951, he developed his version of FORTRAN in order to program an aerodynamics problem for the MIT instrumentation laboratory. As a graduate student at Harvard in mathematics he had been hired part-time by the instrumentation lab to analyze their problems on the only digital computer then in existence at MIT, the Whirlwind II computer of the instrumentation laboratory.

After graduation Neal Zierler went to work for Oliver Selfridge in Davenport's group. He became the Lincoln Laboratory expert on the linear sequence generators of Sol Golomb and their properties. These linear polynomials of delay elements generated digital sequences which had the appearance of being quite random. As a consequence they were important then as now for spread-spectrum applications.

Somewhat after the time of my work on Reed-Muller codes, Neal came to me one day and said, "Irv, I want you to teach me about error-correcting codes."

Since Neal Zierler was an outstanding expert on modern algebra and linear polynomials over the field $GF(2)$ of two elements, I told him that he probably knew already as much as I did. However, I gave him all of the reports and papers on coding in my possession up to that time. Shortly thereafter, he showed me his demonstration that the first-order Reed-Muller codes are equivalent to certain cyclic codes which only recently at that time had been invented by a Dr. Prange at the Air Force Cambridge Research Center (AFCRC). Dr. Zierler used the fact that any cyclic code is encoded by a linear sequence generator.

During the mid 1950s much of my time was spent on the development of automatic digital processors for radar data. My early work on digital machines culminated in 1957 and 1959 with the design of the first all-solid-state (transistor) computer, called CG-24. This machine had many other "firsts" as well: the first machine to be designed and developed using the RTL language, the first computer controlled by a micro-program, the first machine to be emulated in another computer, the first general-purpose machine to have a rudimentary interrupt structure, etc. Finally, as an Associate Group Leader of the largest radar receiver group of Lincoln Laboratory,

I had a number of theoretical physicists and electrical engineers in this group who worked directly for me.

As a part-time hobby I kept returning to the study of modern algebra, keeping in mind the possibility that abstract algebra might help to generalize the Reed-Muller codes I had worked on earlier. Part of the study was to learn all that I could about finite fields, the invention of the early 19^{th} century mathematician, Evariste Galois. Galois was a young French math whiz who developed a theory of finite fields, now known as Galois fields, before being killed in a duel at the ripe "old" age of 21.

For well over 100 years mathematicians looked upon Galois fields as elegant mathematics but of no practical value. It was my idea to use the elements of a finite field as an alphabet to use symbols rather than bits, e.g., half-bytes or bytes for the symbols. This was the beginning of the thought process that led ultimately to the Reed-Solomon codes.

The chance happening that brought Dr. Gustave Solomon into my group occurred in early 1958. My division leader came to me with the dilemma of what to do with a personable young MIT graduate in mathematics, in fact an almost pure algebraist. The physicist-engineer dominated laboratory could not find a place for a person so oriented in theory. With my love of algebra I readily agreed to see how he would do in my group. Actually, I did not have any immediate idea of what I would do with this young algebraist, Gus Solomon, but that coding was uppermost in my mind.

To see how capable Gus was in mathematics I suggested that he generalize what was called, then, the necklace problem: given N black and white beads, how many distinctly different necklaces could be strung? The problem which Gus solved quickly was to count the number of distinct patterns of beads in a rectangular array of $M \times N$ black and white beads, connected as a torus, i.e., shaped as a donut. We named this problem, "the Wampun Problem," for the type of money used by the American Indians, rectangular arrays of colored beads sewn together.

I next showed Solomon my work on coding using Galois fields and the nature of the theorem that needed to be proved. We eventually established this theorem and in 1959 wrote the five page paper, "Polynomial Codes Over Certain Finite Fields," for the *Journal of the Society of Industrial and Applied Mathematics* (SIAM), published June 1960.

Perhaps it was the Galois field incorporation; perhaps it was the nonbinary nature of the code; but for years after its publication the Reed-Solomon code was viewed as interesting mathematics and little else. It simply did not appear to be practical with the computing capability of the day. Even in the mid 60s, when people at JPL began to build and fly spacecraft with error-correction coding, they turned not to the Reed-Solomon code but to the more straightforward but less powerful Reed-Muller code. Such was the case for the next decade. However, in the years since the late 1970s, concurrent with the development of more powerful computers and more efficient decoding architectures, such as that of Berlekamp, the Reed-Solomon code has been widely used in industrial and consumer electronic devices. At present billions of dollars in modern technology, the compact disc memories, digital communications, etc. depend on ideas that stem from the original work of Solomon and myself almost 34 years ago.