JOINT COMPUTER COMMITTEE

SENEWS

SCIENCE EDUCATION SUBCOMMITTEE NEWSLETTER

Vol. 1, No. 2 September, 1958

Purpose

SENEWS is a newsletter addressed to computer oriented members of IRE, ACM, and AIEE to help them promote interest and knowledge among high school age students; it provides a nationwide medium of communication pinpointed to this subject; it evolves from volunteer efforts of the JCC Science Education Subcommittee and relies on its readers for news material.

Write to: C. W. Farr, Chairman, JCC Science Education Subcommittee, M.I.T. Lincoln Laboratory, Lexington 73, Mass.—or to your representative on the *SENEWS* Editorial Board: Richard W. Melville, IRE; George E. Forsythe, ACM; G. L. Hollander, AIEE.

Dangerous Voltage

SENEWS bristles with stories of computer activity in high schools; but *do not* expect your local authorities to welcome you with open arms when you knock on the door and announce your intention to help them "discover" computers. Dr. L. C. Van Atta, a leader in the energetic and successful industry-education program in southern California, has written us: "A word of warning. It has been my experience that school systems tend to regard individual companies, specialized professional groups (petroleum engineers, aeronautical engineers, electrical engineers), and others as pressure groups unless their offers of assistance to the schools are on a very broad basis. Naturally the schools and community are interested in the total education of the student, rather than in biasing the student toward any particular specialization."

Dr. Van Atta went on to point out that innovations (no matter how good) represent a disruption to established curricula. Since Sputnik the well-meaning offers of assistance have multiplied. In Los Angeles the superintendent of schools has created a central committee for cooperation, and the school system has provided a full time Executive Secretary to coordinate the community efforts and "arrive at a broad program in which al specialized subjects have their appropriate emphasis."

Avoid burning up the circuit; measure the input impedance of the school system in your community before applying your driving voltage.

Have Design, Will Build—Chapter Two

In the last issue we reported how David Ecklein, a high school junior from Cedar Falls, Iowa, was building a checker playing digital computer at home, using his own design, thirty-five hundred surplus tubes, and the help of his buddies in Tom Sawyer style. (*Ed. note:* Last minute correction of number of tubes was not in time for all publications.)

David's project has been interrupted for a while by summer employment at IBM's research laboratory in Poughkeepsie, N. Y., where he will learn to program and operate one of the major computers. He is looking forward to talking with engineers about circuit designs, and learning about the well-known checker playing program of Dr. A. L. Samuel. Nobody who has met 17-year old David will be greatly surprised if he makes some significant improvements in the techniques before the summer is out.

David Ecklein's opportunity to work in a major computer laboratory came about through his contact with the JCC Science Education Subcommittee. It is, in David's words, "beyond my fondest hopes. Not only will it provide experience but also a means to earn funds to carry out my project to conclusion."

Werner Buchholz

Junior High School Computer Programmers

The thirteen year old in your home may not have learned to write the following coded program for the IBM 704 computer to solve (for 5 values each of a, b, and c) the equation $P=(a+b)(a^2+ab+b^2)(c+1)$. But don't be discouraged. We know of only one Junior High which has provided such instruction.

TABLE I

Symbol	Operation	Address, Tag, Decrement	Remarks
SBM	LXA	*HERE, 1	Load index register
	CLA	*A DATA+5, 1	$(a+b)$
	FAD	*B DATA+5, 1	
	STO	*TEMP	
	LDQ	A DATA+5, 1	a^2
	FMP	A DATA+5, 1	
	STO	TEMP+1	
	LDQ	A DATA+5, 1	ab
	FMP	B DATA+5, 1	
	STO	TEMP+2	
	LDQ	B DATA+5, 1	b^2
	FMP	B DATA+5, 1	
	STO	TEMP+3	
	CLA	TEMP+1	(a^2+ab+b^2)
	FAD	TEMP+2	
	FAD	TEMP+3	
	STO	TEMP+4	
	CLA	*C DATA+5, 1	$(c+1)$
	FAD	*INFO	
	STO	TEMP+5	
	LDQ	TEMP	$(a+b)(a^2+ab+b^2)=$(PROD.)
	FMP	TEMP+4	
	STO	TEMP+6	
	LDQ	TEMP+5	(PROD.)$(c+1)=$(ANS.)
	FMP	TEMP+6	
	STO	*X INFO+5, 1	
	TIX	SBM+1, 1, 1	If contents of index register are greater than decrement (1), decrease contents by decrement (1) and transfer to SBM+1. Otherwise proceed to next instruction (HTR)
	HTR	SBM	Halt and return to starting position.

* Note: 5 is in location 5 HERE; 5 values of "a" start at location A DATA, 5 b's start at B DATA, 5 c's start at C DATA: number 1 is in location INFO; TEMP is a temporary storage register; X INFO is storage location of final result.

Harley Tillitt of the U. S. Naval Ordnance Test Station, China Lake, Calif., conducted an educational experiment with 24 selected eighth grade students at Burroughs Junior High School in November, 1957. The experiment involved lectures and demonstrations during 10 forty-minute instruction periods. Computer programmers will recognize the conventions developed by SHARE, an informal programming organization of IBM 704 users.

"The success of the students in these experiments shows that it is feasible to introduce computer programming instruction below the college level," concluded Tillitt in the report submitted to *SENEWS.* The problem coded here (Table I) is one of 14 problems assigned to eighth grade students during the experiment. The complete report came to *SENEWS* via George Forsythe. (*Ed. note:* If this report is published in more detail, *SENEWS* will announce it in a later issue.)

SADSAC II

Samson Additive Digital Sequential Automatic Computer, SADSAC II, carried its designer-builder, Peter Samson, to first place in the Massachusetts State Science Fair, and fourth place in the 1958 National Science Fair. Peter lives in Lowell, Mass., and will enter M.I.T. this autumn with a four year National Merit Award scholarship. He visited Lincoln Laboratory recently on invitation from the Science Education Subcommittee—and when he left it was *we* who were inspired.

First we reviewed specifications. SADSAC II is a relay computer, designed to add, subtract, and complement four bit binary numbers; input is from two punched paper tape readers, and programming provides for jumping from one tape to the other; output is a solenoid actuated typewriter (actually a home rigged 1898 Oliver).

But then we explored motivation, and the "engineering economics" of student computers. Peter's early science interest found roots in his father's electrical (hi-fi, etc.) equipment. High school algebra whetted his mathematics appetite. Attempting to build a machine to play ticktacktoe led him to design his own arithmetic circuitry, and the home grown SADSAC I computer was the inevitable result. After winning local honors with SADSAC I he went on to national fame with its successor.

Peter did not have help from a computer expert advisor; like most student builders of computers he dug the know-how out for himself, "and used common sense," he hastens to add. He regarded pertinent literature as scarce but good; the hardware situation he found pretty sad. We asked why he used relays instead of tubes or transistors. He pointed out that relays are cheap and are ac energized at a few standardized voltages; tubes and transistors involve not only more money, but more complex circuitry and power supplies. He used discarded pinball machine components, and reliability was a nightmare.

"Building SADSAC III this summer, Peter?"

"No. I want to get in some study that I can't find time for during the school year."

Book and Film

"Local action kit," prepared by the President's Committee on Scientists and Engineers, Washington 25, D.C.

This kit is designed to: 1) tell how to bring together community organizations with the same basic objective into a single program, 2) promote self-evaluation and the recording of successful techniques and experiences useful to other communities throughout the nation, and 3) establish a pattern of nationwide effort without sacrificing the incentives and benefits of creative local action.

Here is what the kit contains: 1) "Guidebook for local action," suggesting general techniques adaptable

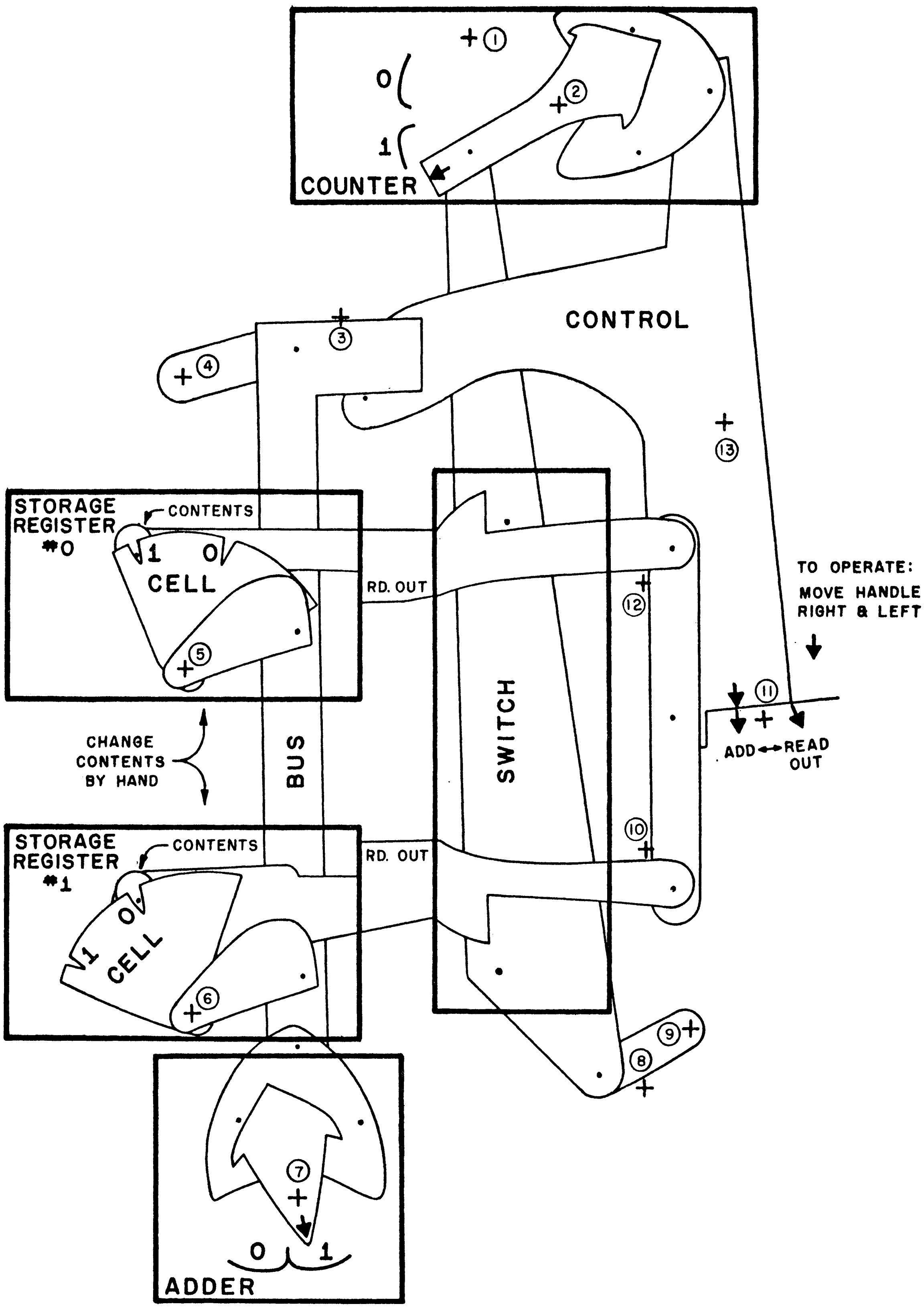

Fig. 1—PAPAC-00, a 2-register, 1-bit, fixed-instruction binary digital computer (Rollin P. Mayer, July 14, 1958).

to the needs and resources of your own community, 2) examples of tested projects in other communities, relating how they began, how they are being carried out, their scope, financing, and results, 3) reference materials such as information on scholarships, improvement of science curricula, youth activities, etc., and 4) a bibliography of selected materials, visual aids, organizations, and publications helpful in your community program.

This kit was prepared for use in local programs for the improvement of science and mathematics education (not limited to computer education). It is available without charge to *groups*, not (for budgetary reasons) to individuals.

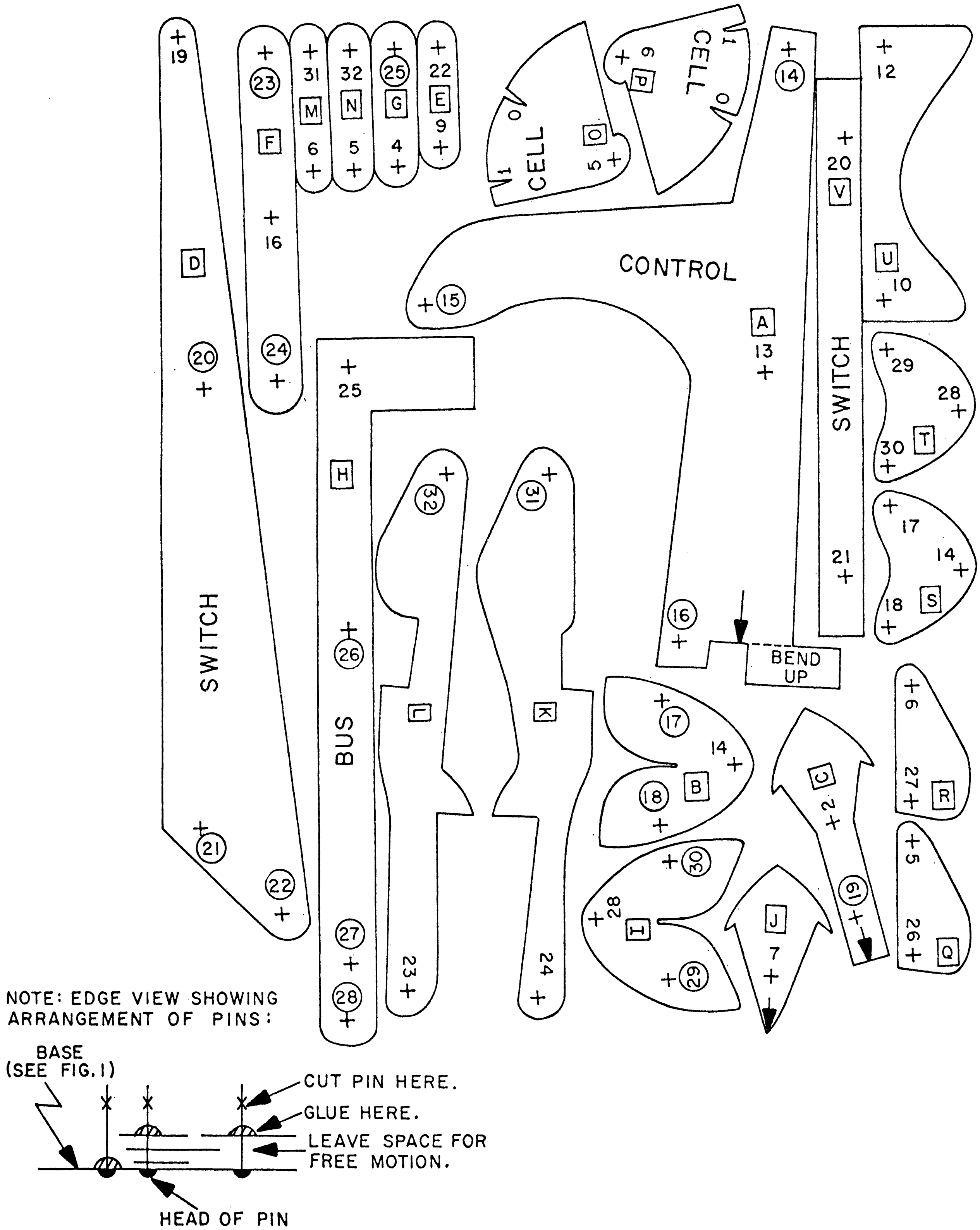

Fig. 2—Parts for PAPAC-00 (Rollin P. Mayer, July 14, 1958).

Interested individuals can get free copies of "Local action," the monthly news bulletin. (*Ed. note:* Don't be misled by the price; this is good stuff.)

PAPAC-00, A DO-IT-YOURSELF PAPER COMPUTER

In less than an hour you can build the simplified digital computer shown in Fig. 1, using only a pair of scissors, three dozen common pins, and the parts shown in Figs. 1 and 2. This computer was developed from the model demonstrated in the Concord, Mass., High School lectures on computers reported in the first issue of *SENEWS*.[1]

From the discussion below, the computer expert will

[1] *SENEWS*, IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-7, pp. 186–187; June, 1958.

recognize that "PAPAC double zero" contains most of the units of a large-scale computer, but in simplified form. The *control unit* includes a counter and a system for controlling the parts of the computer according to the instruction being performed (in this model a simple fixed instruction is used; a large computer can draw from several instructions obtained from storage). The *storage unit* includes registers, bus, and selection switch; register contents are changed by hand rather than by the computer. The *arithmetic unit* can add. *Input and output units* have been eliminated by allowing the operator to deal with the insides of the computer directly rather than by way of complicated equipment. Proprietary rights are held by the author.

In *operation*, PAPAC-00 follows the same fixed instruction over and over again. This instruction is: "Read the number out of the currently-selected storage register and add it to the adder, then get ready to use the next storage register for the next time." The "counter" keeps track of which storage register to use next; since there are only two registers, numbered "0" and "1," the counter alternates between them. The "switch" is controlled by the counter and allows only the selected register to be operated. Each "storage register" contains only a single binary "cell"; when the register is operated, the cell is forced against the "bus" if the cell is set to "1." If a "1" has been read out in this way, the bus actuates the "adder," preparing it to add the "1." If the cell is set to "0," the bus and adder are not operated, and "0" is added to the adder. Binary sums are as follows: $0+0=0$, $0+1=1$, $1+0=1$, $1+1=10$. The adder forms these sums correctly except that in the last case it forms a sum of "0" because it can handle only one digit The "control," pushed back and forth by hand, performs this fixed instruction by operating the counter and switch, and by returning the bus to its "0" position (if it had read out a "1") causing the sum to be formed in the adder.

To *assemble* PAPAC-00, Fig. 1 should be used as the base, and the shapes of Fig. 2 should be fitted over it by following these steps:

1) Punch a pinhole exactly through the intersection of each cross (+) in Figs. 1 and 2 (but not the dots in Fig 1).

2) Cut out exactly on the lines, the parts in Fig. 2, in any order They are marked with a letter in a square box, from [A] to [V], and the next steps will be easier if you place each piece on the table in alphabetic order as you cut it out.

3) Place a pin up through each hole with a circled number (from (1) to (32)).

4) Taking each part of Fig. 2 in alphabetic order, place its *un*circled number holes down over the correspondingly numbered pins.

5) In first operating the computer you may find that some parts jam because the upper piece is down too far on the pins: pry such pieces up a little to provide space for free motion.

6) The construction can be refined by cutting the pins and gluing the uppermost part to the remaining length. Caution:

1) Don't cut the stop pins too short.

2) Glue only *one* moving part to the same pin.

Rollin P. Mayer