The Personal Charites (PC)

Combining the powers of Aglaea, Euphrosyne and Thalia

by Zeus the Almighty
most creative God of all

616 B.c.

— 1 —

THE PC

This is a prototype for a a future invention called the "personal charites" (PC). It is to be presented to all human beings in approx. 2000-3000 years and will turn the human world into a new era. This prototype however will remain in the hands of the gods as it is too powerful to be handled by mortals. Instead, we will slowly introduce a simplified version --- with bits instead of trits --- starting around the 1800s. Ada and Charles are going to be our first test subjects, and from there we will see.

We outline a different prototype for internal use only. This is called a "ternary central processing unit": a TCPU or 3-CPU. This machine is gifted with the powers of the Graces/Charites:

0. A: Aglaea, the grace of festive radiance;

1. E: Euphrosyne, the grace of good cheer;

2. T: Thalia, the grace of abundance.

Each trit can be encoded using the unbalanced encoding (0, 1, 2) or the true encoding of A, E, T. 9 trits combined make for a Trix (plural: trex) and can be represented as 3 base-27 digits in the $0-9a-q$ alphabet, much like the unknown 2x$0-9a-f$ hexadecimal alphabet. Finally, every trit and trix as a numeral value is right-side significant, meaning numbers are read from right to left. As an example, $AT = 6_{dec}$ and $TA = T = 2_{dec}$. Similarly, a trix of $q00 = q = 26_{dec}$, and $110 = 11 = 28_{dec}$.

Since a trix contains a high volume of information ($UTRIX_{MAX} = 3^9 - 1 = 19682$), every register holds 1 trix.

36 trits combined (4 trex) represent an "instruction", an atomic piece of information describing one simple thing it needs to do. After this instruction is done, it will automatically continue with the next instruction. This is not logic, do not worry. It is just magic.

I do not need to remind you that Aglaea, Euphrosyne and Thalia work in mysterious ways and although my attempts to control them here seem fruitful, they never cease to surprise.[1]

[1] May the god Khepri be with us

— 2 —

QUICKSTART

Together with this document, one will find a round object called a wheel. This will needs to be installed in a "virtual environment" or "venv", like this:

```
apt install python3-venv
python3 -m venv venv
. venv/bin/activate && pip install ./*.whl
```

After this, one can activate the virtual environment using ". venv/bin/activate". This exposes a new program: "PCharites". To execute instructions in PCharites, you can use either "file" or "exec" directly. Afterwards, you have to specify whether the instrucitons are encoded in "TRIT" or "TRIX" -- more on this later.

EXAMPLES

Performing $l + l \to reg_1$ $(21 + 21 \to register\ 1)$ is performed with the following TRIT ins: $ATTAAATTEAETAAAAAAAETAAAAAAE$. To execute this, we can run the following command:

```
PCharites exec TRIT ATTAAATTEAETAAAAAAAETAAAAAAEAAAAAAAA
```

This should give the following output:

```
ERROR: Invalid PC 1
```

Since instruction 0 was executed but the computer was not stopped with a STOP instruction. To gracefully stop, add the stop "ETAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE".

```
PCharites exec TRIT ATTAAATTEAETAAAAAAAETAAAAAAEAAAAAAAA
ETAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
```

When programs become bigger, the TRIX representation might become easier to use:

```
PCharites exec TRIX o0hl00l001007ddddddddddd
```

You can also write this to a file and then run it:

```
echo 'o0hl00l001007ddddddddddd' > my_program
PCharites file TRIX ./my_program
```

or

```
echo 'o0hl00l001007ddddddddddd' > my_program
cat my_program | PCharites file TRIX -
```

NOTE: ALTHOUGH IT MAY BE TEMPTING, PLEASE DONT REVERSE THE WHEEL/PYTHON PACKAGE. THAT IS NOT INTENDED.

— 3 —

THE INSTRUCTION

The instruction is built up from 36 trits or 4 trex. In general, an instruction is divided by trex. The first trix contains the opcode and other metadata about the instruction. The second trix contains the first source. The third trix contains the second source. Finally, the last trix contains the destination. The last three trits of the opcode encodes the type of each subsequent trex: whether it is a (T) constant; a (E) register or a (A) register pointing to a memory address (reference). Memory is indexed at every 3 trits because. Constant address memory access is not allowed.

The Opcode

Trit 1 will tell us if it is an arithmetic instruction. A as first trit means the instruction is arithmetic positive, for example addition. T performs the arithmetic opposite: where Aglaea adds, Thalia subtracts. Euphrosyne is not bothered by this feud and will instead perform other types of operations.

The ALU: Basics

When arithmetic operations are chosen, the subsequent choice will determine whether it is a trit-move operation (A), a ternary-logic operation (E) or a numerical operation (T). The next trit will decide on the sub-type:

Figure 3.1: Instruction breakdown

A	AA	trit-mov: roll right
A	AT	trit-mov: shift right
A	EA	logic: OR
A	EE	logic: AND
A	ET	logic: NOT
A	TT	calc: add
A	TE	calc: multiplication
A	TA	calc: pow
T	AA	neg trit-mov: roll left
T	AT	neg trit-mov: shift left
T	EA	neg logic: NOR
T	EE	neg logic: NAND
T	ET	neg logic: NOTNOT
T	TT	neg calc: subtract
T	TE	neg calc: division
T	TA	neg calc: sqrt

Jump around

If *E*uphrosyne is chosen at first, choosing Aglaea will make one jumpy. How jumpy one becomes depends on your next choices. The following tabulars have been found to be helpful in understanding their behaviour:

EA	A	jmp
EA	E	call
EA	T	ret

(a) index 2

EA.	A	relative
EA.	E	absolute
EA.	T	-relative

(b) index 3

EA..	A.	if ZERO
EA..	E.	always
EA..	T.	if PARITY
EA..	.A	if OVERFLOW
EA..	.E	always
EA..	.T	if CARRY

(c) index 4, 5

Source 1 will be the input (except with RET), source 2 and the destination are unused.

Calm Euphrosyne

Choosing Euphrosyne twice will make the computer very calm, nopping its way through life. If instead Thalia is chosen, weird things may happen.

ET	T	INTERRUPT k4
ET	A	STOP
ET	ATTT	DSTOP
ET	EA	POP
ET	ET	PUSH

— 4 —

K4 INTERRUPTS

When a program is interrupted, the three ladies will be systematically called together so serve you. They consider your r0-input as command and will look up what to do with this. If necessary, r1-r5 will serve as input registers whilst r6- will remain virtually untouched. So far, the following r0-inputs have been found to respond:

r0 Trix	r0 Trit	Description
0	A	State reset
1	E	write r1 trix to stdout
2	T	Write r1 trit to stdout
3	AE	Write [r1] ASCII to stdout where possible
i	AAT	Write userinput to [r1]
b1	TAEE	execve(r1, r2, r3)
o1	ATTE	ls [r1], write to stdout
p1	ETTE	cat [r1], write to stdout
..	..	..
k4	TATEE	SMITE FLAG0 to stdout
1f2	EAAATET	Write FLAG1 to stack