Solder-Defined Computers for Provable Immunity Against Hacking and Malware

Marc W. Abel

Department of Computer Science and Engineering

11 May 2023

WRIGHT STATE
UNIVERSITY

Three walls to defend

- Software
- Personnel
- Hardware

Four kinds of hardware problems

- Outdated approaches ignore security
- Excessive complexity hides problems
- Manufacturer interests prevail
- Silicon chips can't be repaired later

Three freedoms sought

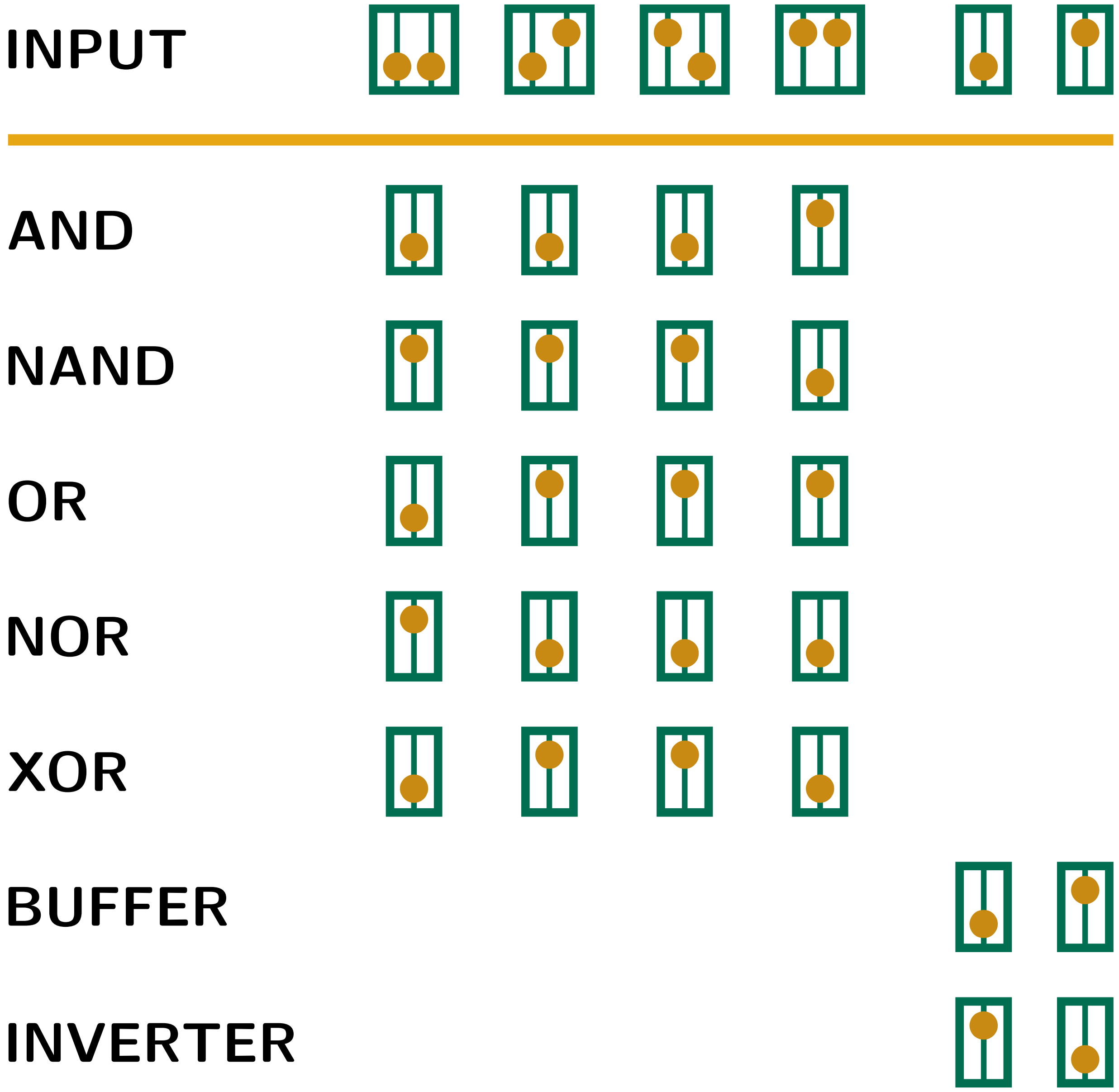
- Independence from vendors
- Full ownership rights
- Permanent security

Two enablers of success

- Surface-mount technology
- Firmware in RAM as logic

WRIGHT STATE
UNIVERSITY

Seven Basic Logic Gates



INPUT

AND

NAND

OR

NOR

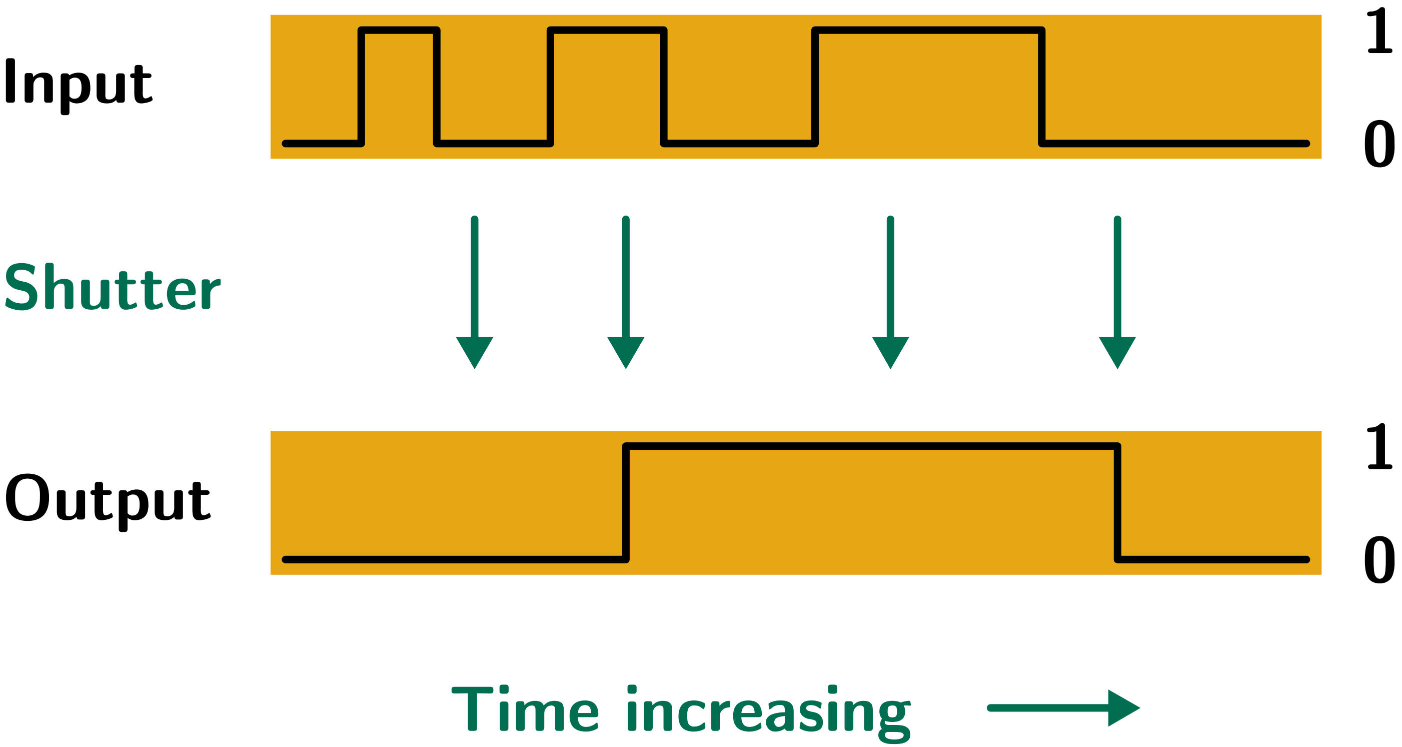
XOR

BUFFER

INVERTER

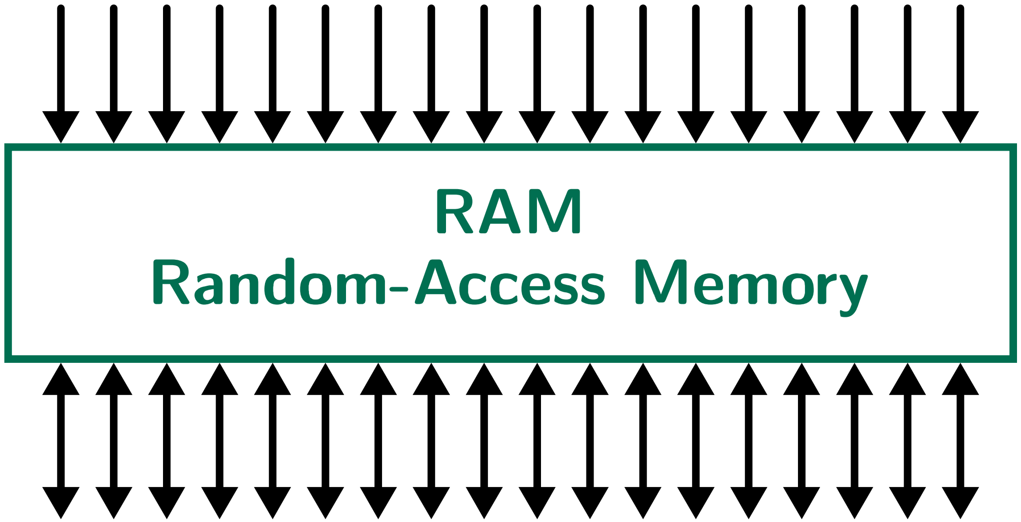
A D flip-flop only changes its output when:

1. told it's time to check, and
2. output doesn't already reflect the input.



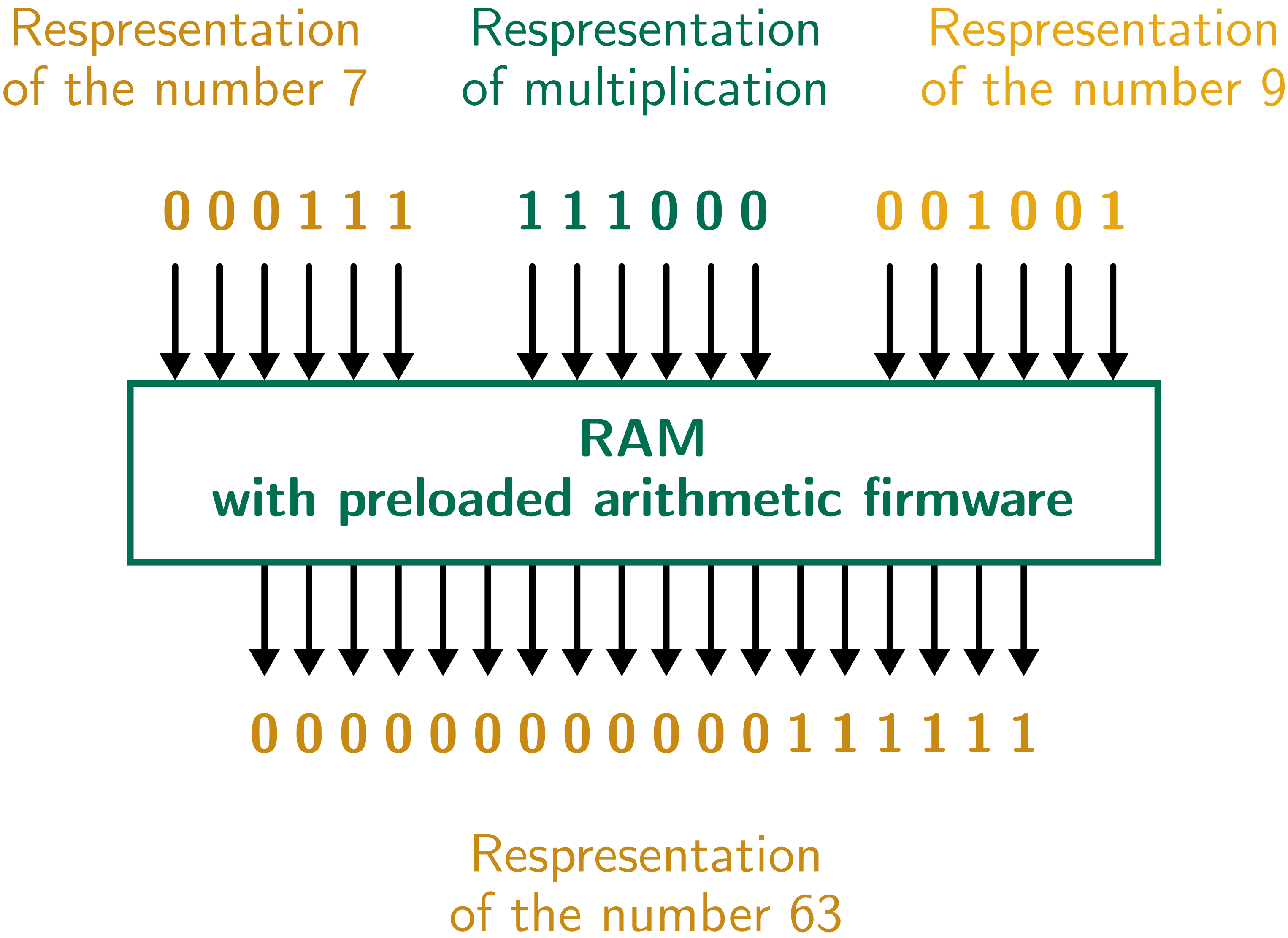
A RAM can remember a lot of 18-bit words.

18-bit "address" where store or retrieve will occur



RAM
Random-Access Memory

18-bit word to store to or retrieve from the given address



Respresentation of the number 7

Respresentation of multiplication

Respresentation of the number 9

0 0 0 1 1 1

1 1 1 0 0 0

0 0 1 0 0 1

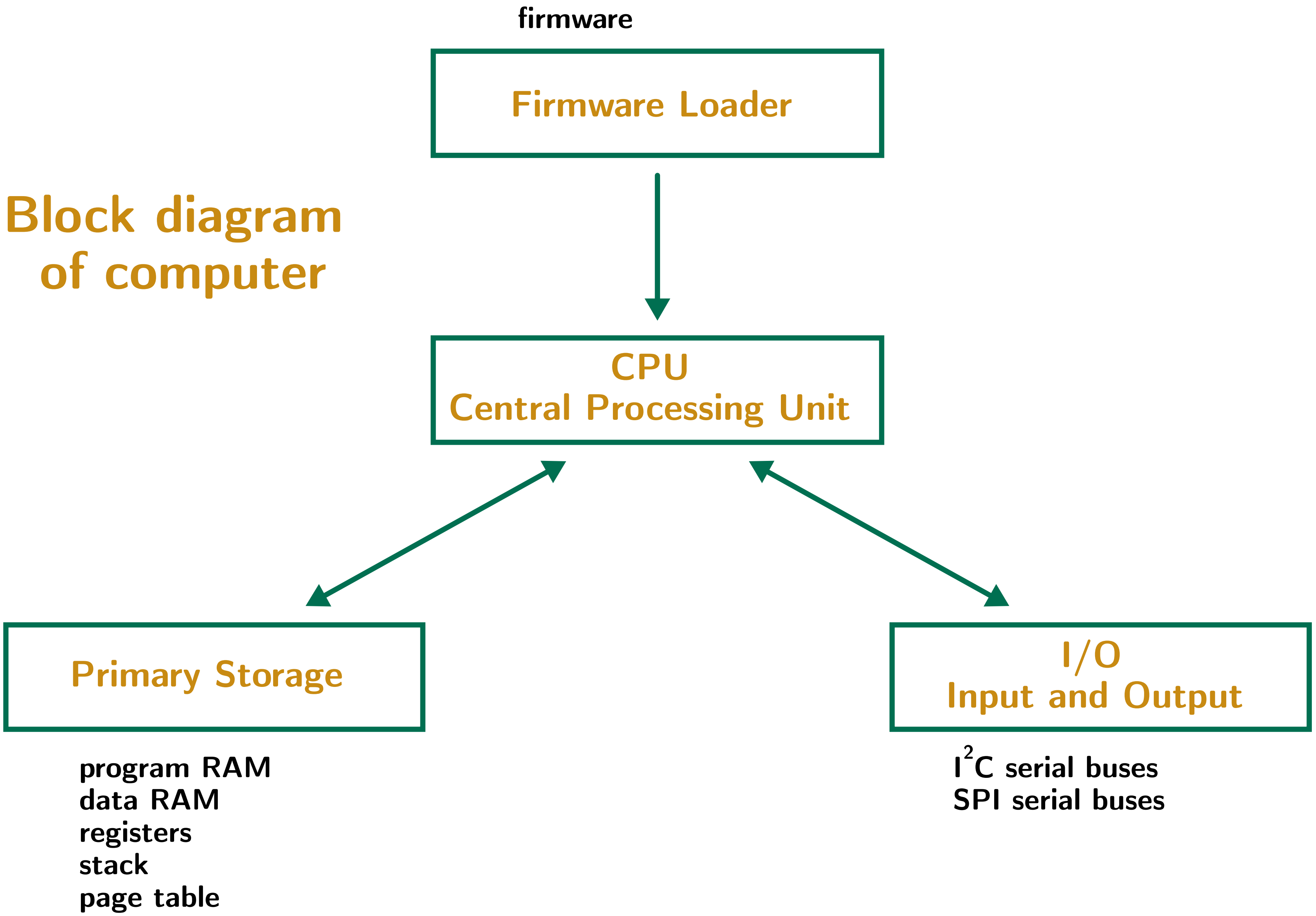
RAM

with preloaded arithmetic firmware

0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1

Respresentation of the number 63

Block diagram of computer



firmware

Firmware Loader

CPU
Central Processing Unit

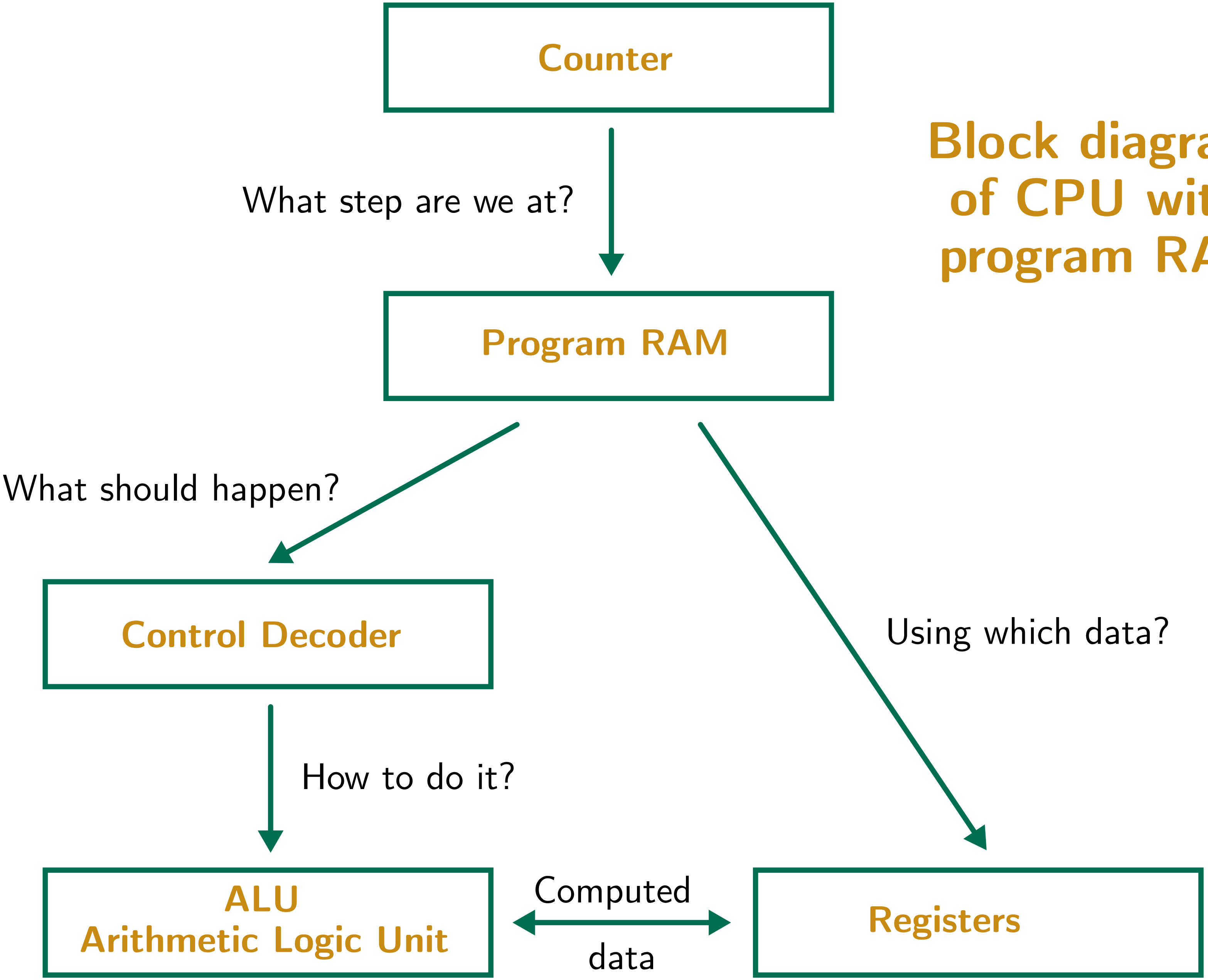
Primary Storage

program RAM
data RAM
registers
stack
page table

I/O
Input and Output

I^2C serial buses
SPI serial buses

Block diagram of CPU with program RAM



Counter

What step are we at?

Program RAM

What should happen?

Using which data?

Control Decoder

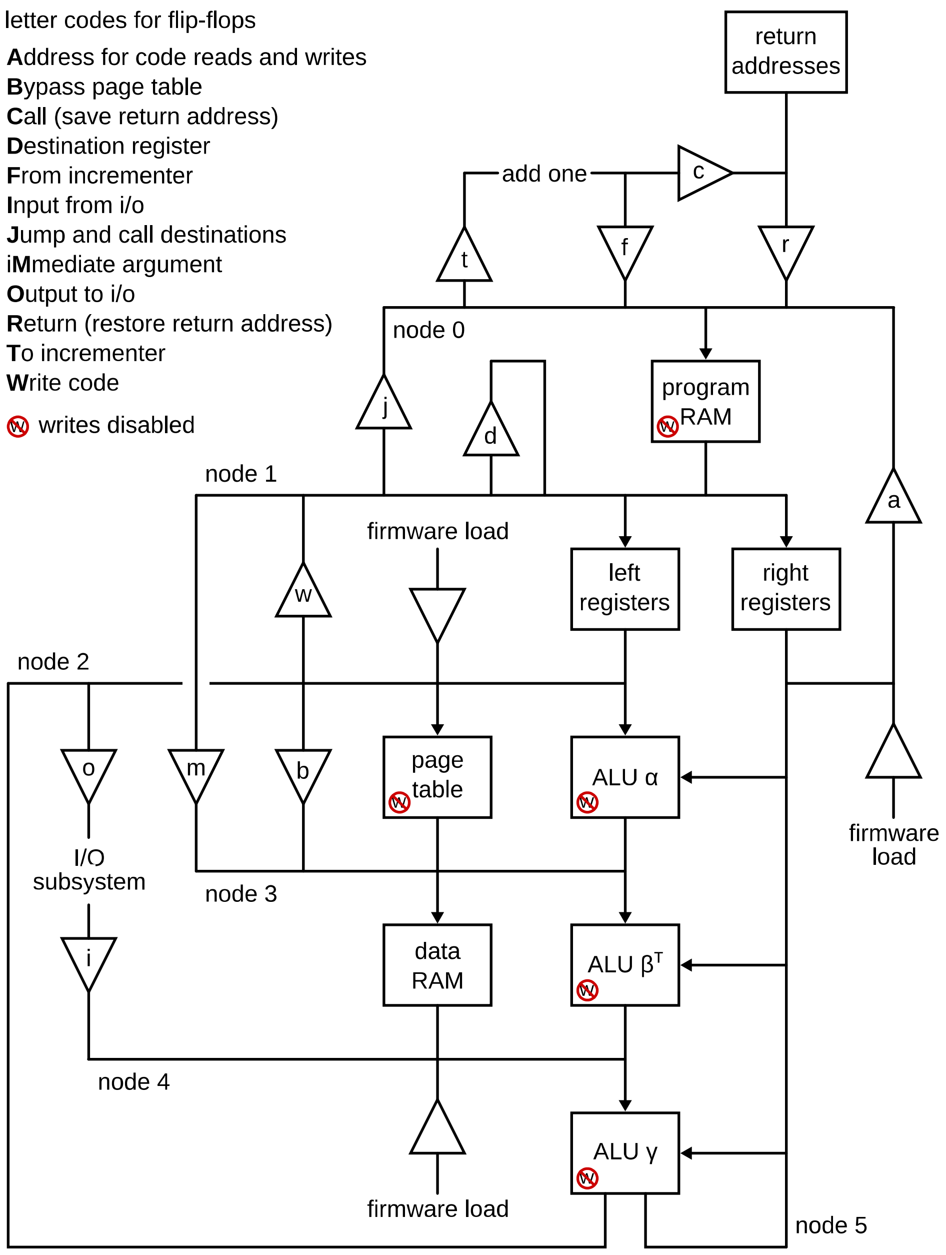
How to do it?

ALU
Arithmetic Logic Unit

Computed data

Registers

CPU Principal Data Paths



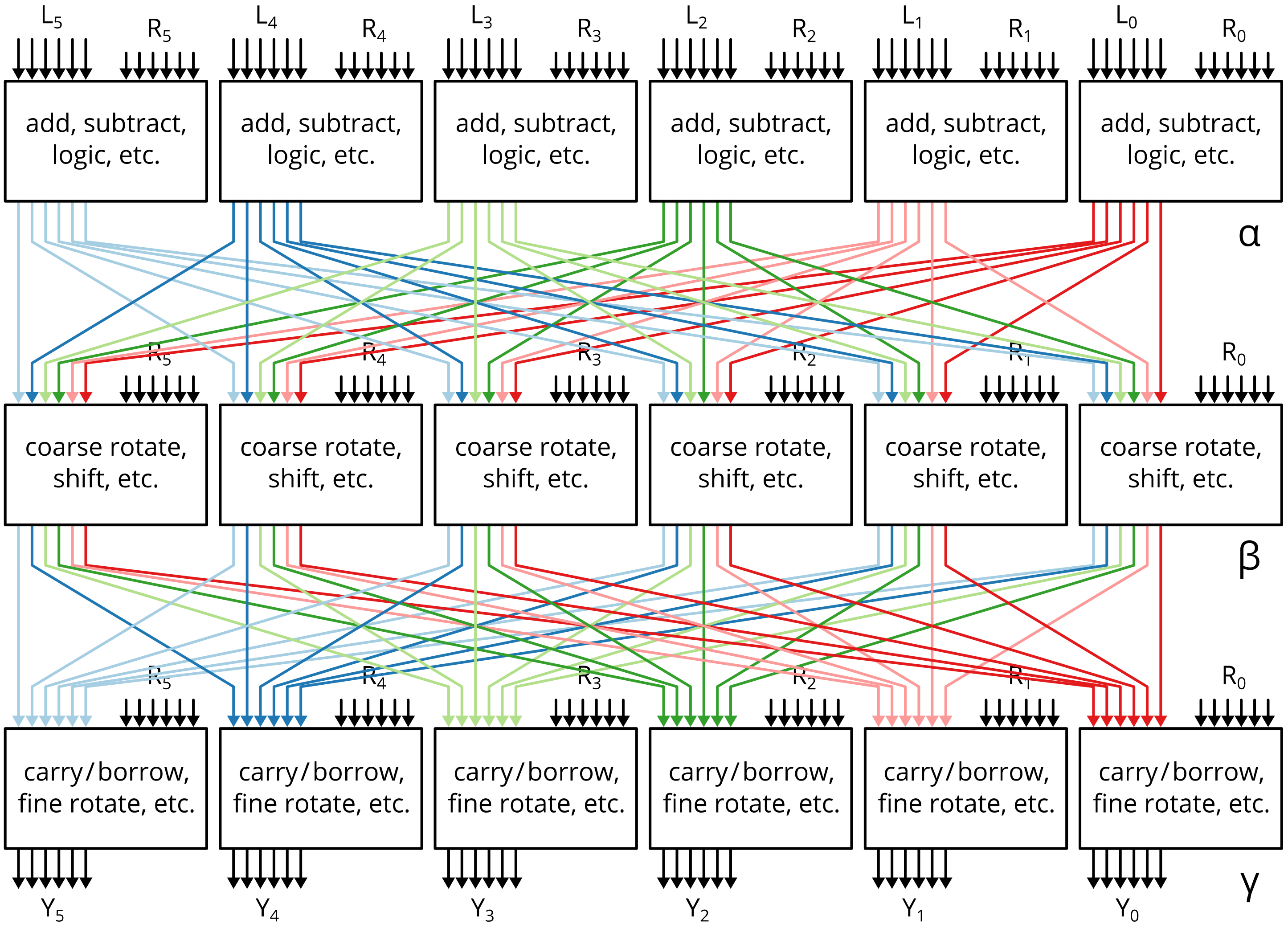
letter codes for flip-flops

Address for code reads and writes
Bypass page table
Call (save return address)
Destination register
From incrementer
Input from i/o
Jump and call destinations
i**M**mediate argument
Output to i/o
Return (restore return address)
To incrementer
Write code

ⓦ writes disabled

Data Layers of ALU

L and R: 36-bit Left and Right operands, 6 bits per slice

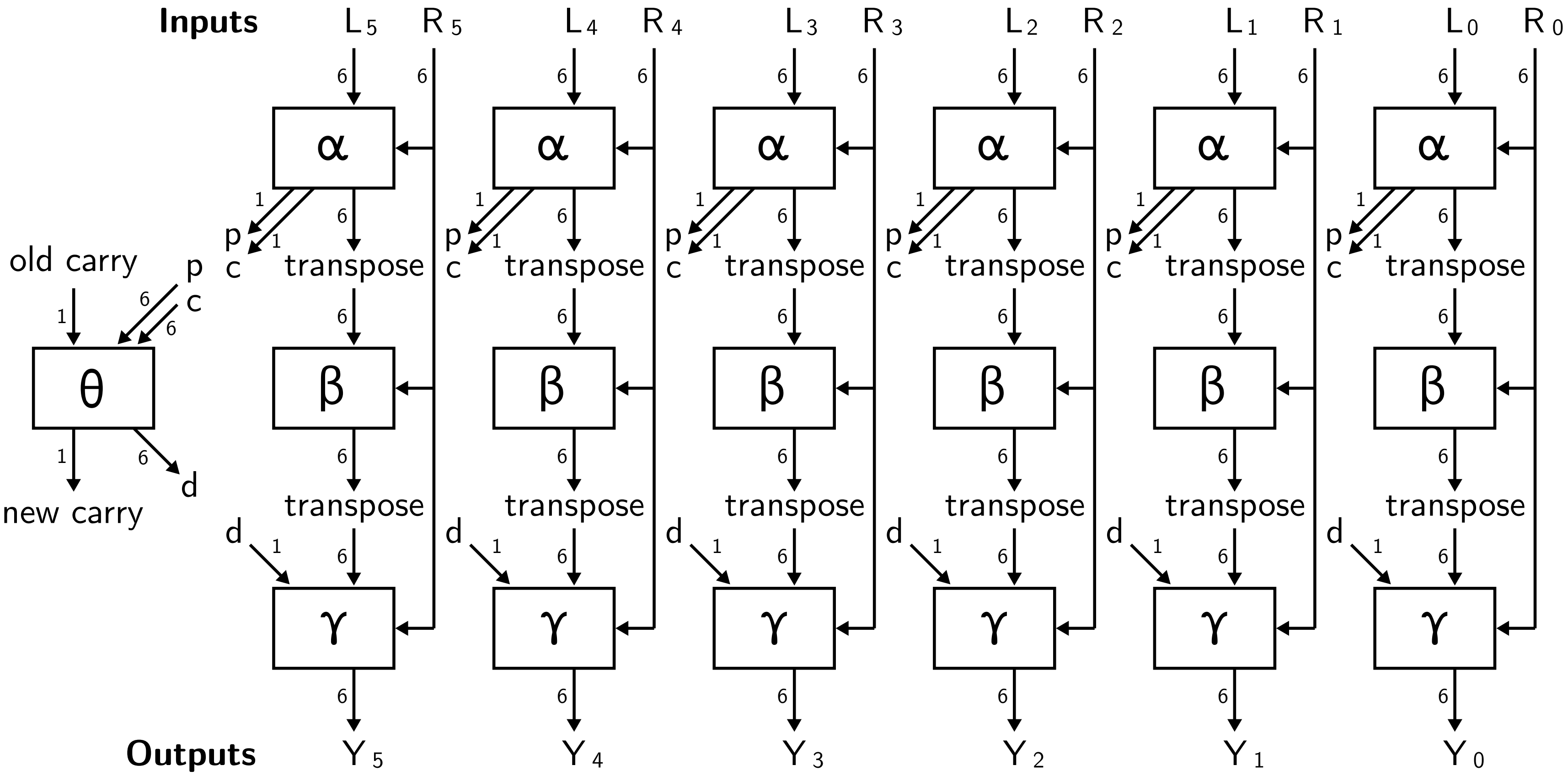


add, subtract, logic, etc.

coarse rotate, shift, etc.

carry/borrow, fine rotate, etc.

Y: 36-bit result, 6 bits per slice

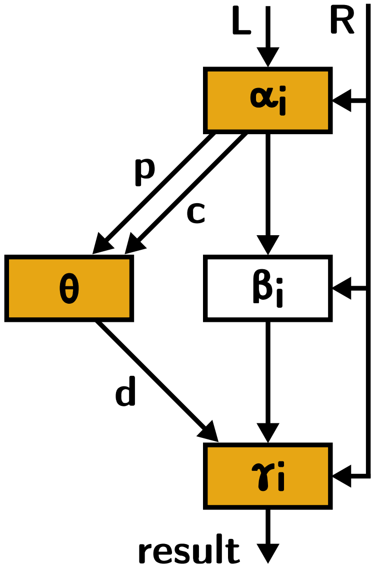


ALU with carry propagation elements shown

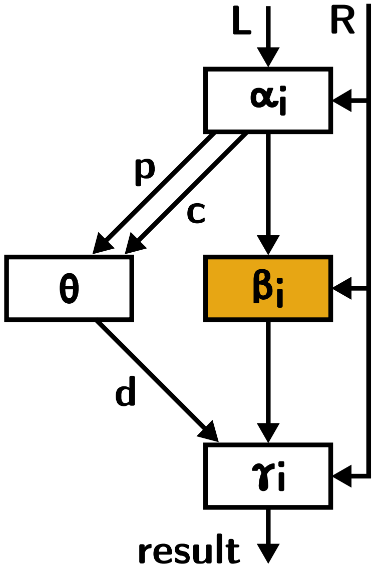
Small digits that are not subscripts indicate number of wires.

Superposition of ALU Operations

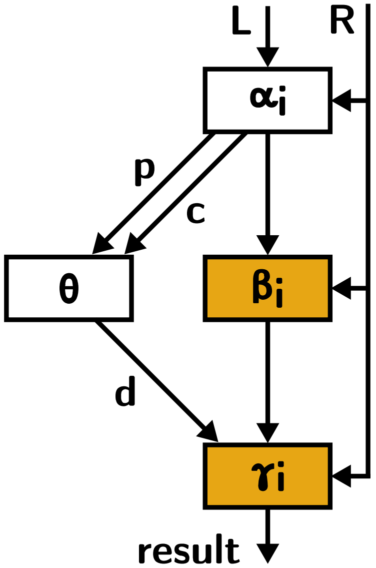
Carry-skip adder



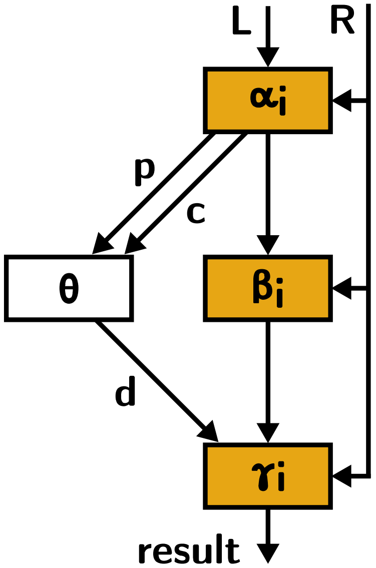
Swizzler



Logarithmic shifter



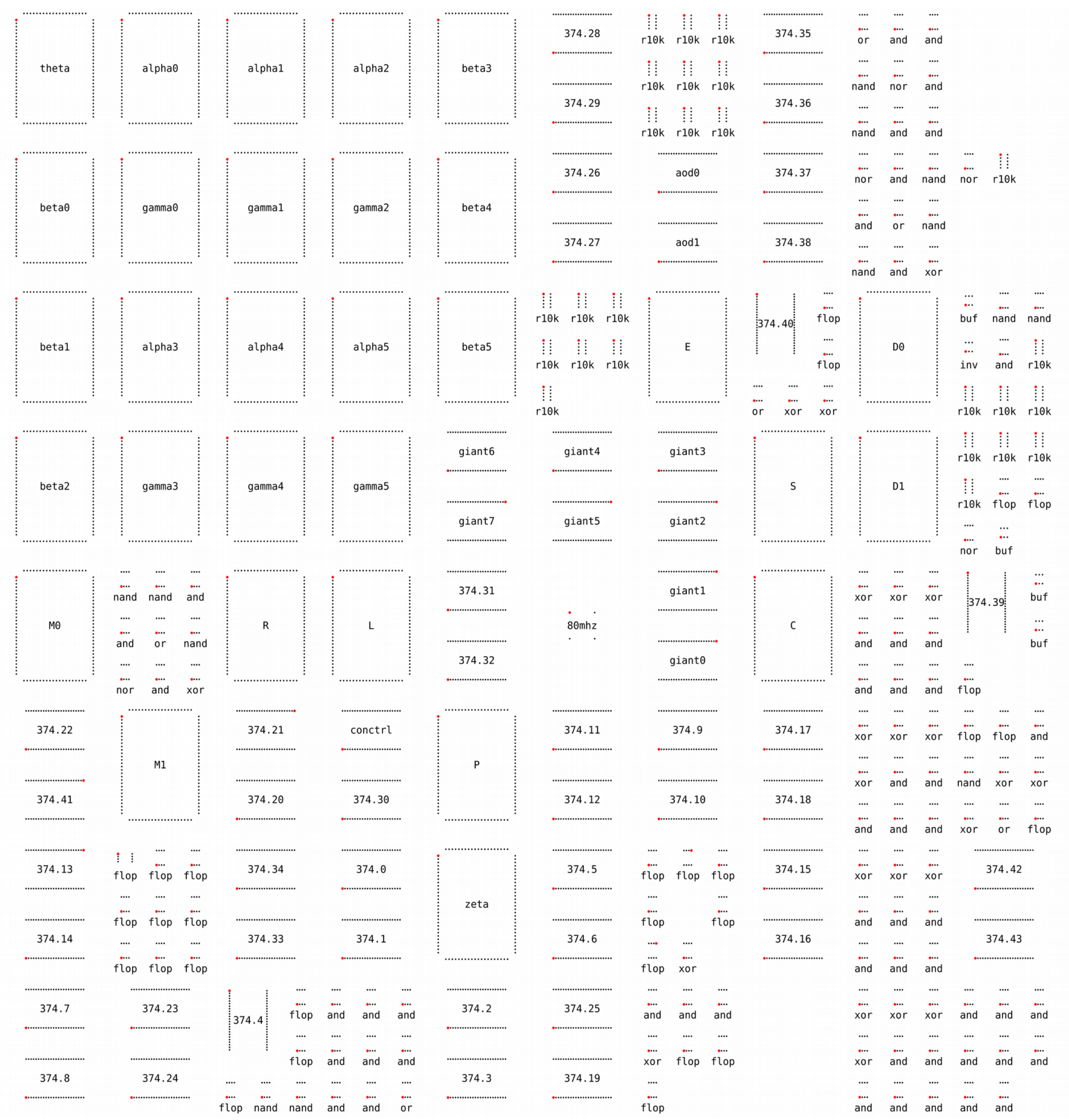
Substitute & permute



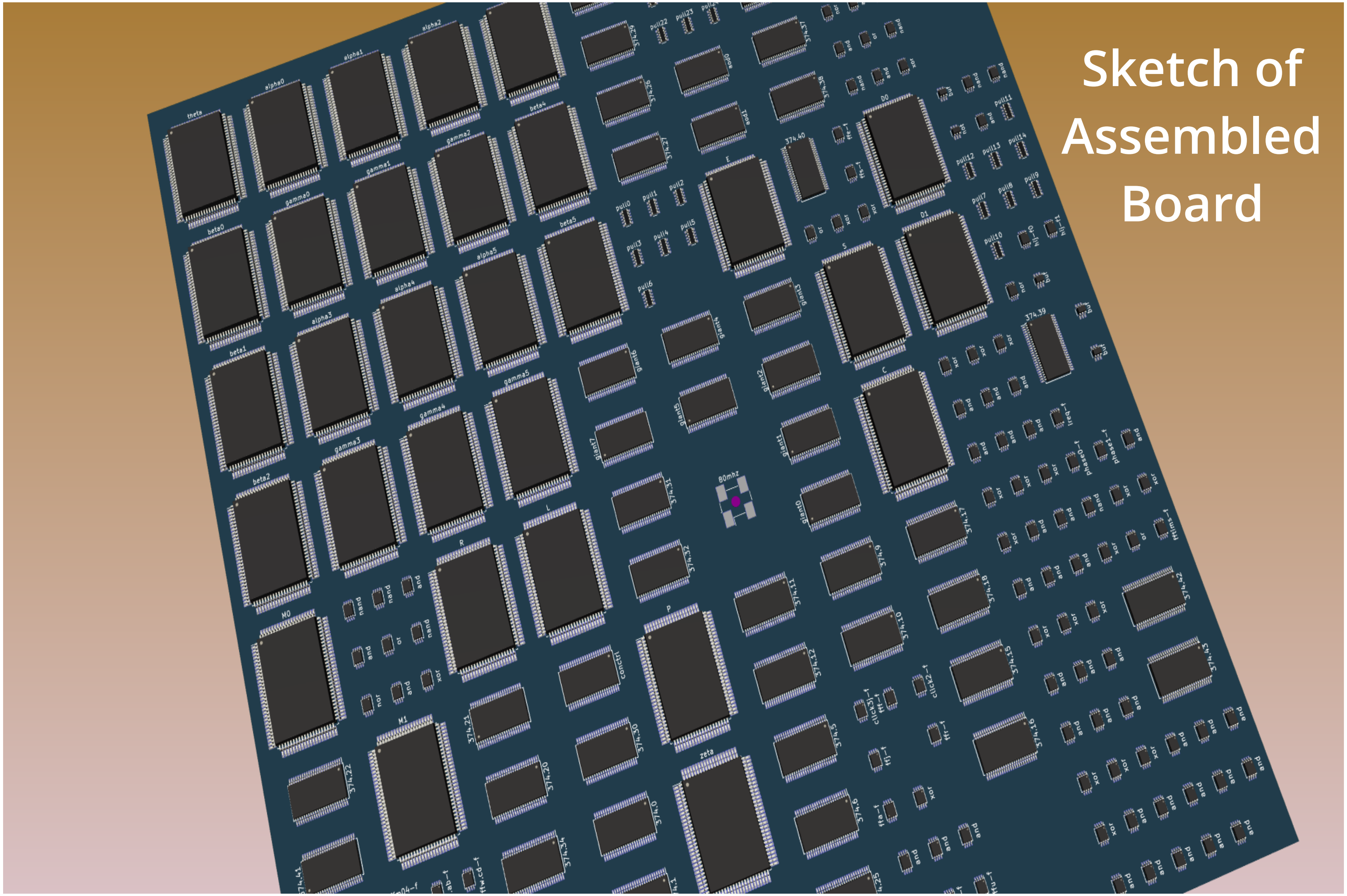
Same circuit
Same chips
Same board space

Circuit Board Floorplan

as of 11 May 2023



Sketch of Assembled Board



Fast Enough For

- Hardened desktop apps
- Electronic mail
- Light- to moderate-use servers
- Controlling objects that move
- Process controls
- Peripheral & device controllers
- Telephony
- Modest Ethernet switches

Too Slow For

- Most Web surfing
- Machine learning
- Image and video processing
- Self-driving vehicles
- Fast raster or vector graphics
- Fast symmetric cryptography
- Fast asymmetric cryptography
- Bioinformatics

Security Improvements

- No vendor lock-in
- No secret functionality
- No purpose of use limitations
- No right to repair infringements
- No privilege escalation via the CPU
- No license fees to build, use, or modify
- Sticky out-of-range flag for all arithmetic
- No encrypted or closed-source firmware
- No DRAM or DRAM-associated vulnerabilities
- Every I/O device confined to its own bus and buffer
- No CPU persistent state except for one firmware IC
- No complex logic from IC manufacturers within CPU
- No program access to stack except CALL and RETURN
- Stack overflow unlikely, can't lead to privilege escalation
- No branch to addresses not present in the instruction word
- Mixed-sign variants for add, subtract, multiply, shift, abs. value

Before This Can Be Built

- I/O subsystem to support SPI and I^2C buses
- Firmware loader
- Resolution of clock skew concern

Ways to Get Involved

- Firmware upgrade for faster multiplication
- Support for integer division
- Floating point like IEEE 754-2019, but 36- and 72-bit formats
- Floating point for compatibility (32- and 64-bit formats)
- More assembler features
- Lightweight operating system
- Lightweight scripting language
- Lightweight programming language
- Minimalist toolchain that can be audited
- I/O device drivers
- TCP/IP stack
- TLS 1.3
- New block cipher to leverage architecture
- Formal verification (similar to seL4 or INTEGRITY-178B)

WRIGHT STATE
UNIVERSITY

https://people.wright.edu/marc.abel