Carrying out instruction: recap

- instruction
- memory
- acc
- addr
- retrieve
- op
- logic
- control
- store
The essence of memory

Have two memory cells: A, address 0; B, address 1. Have a control bit, C (specifies address 0 or 1). What formula describes memory retrieval?

A. not (C and A) and not (not C and B)
B. (C and A) or (not C and B)
C. (not C and A) or (C and B)
D. (C and not A) or (C and B)
E. (not C or A) and (C or not B)

The essence of logic

Have two operations: or (op 0), and (op 1). Given inputs A, B and op code C, what Boolean formula specifies the result?

A. (not C or A or B) and (C and A and B)
B. (not C and (A or B)) and (C and (A and B))
C. (not C and (A or B)) or (C and (A and B))
D. (A and B) or (A or B)
The essence of storage

Have two memory cells: A, address 0; B, address 1.
Have a control bit, C (specifies address 0 or 1).
Have new bit to write W.
What formula in terms of these values describes new value at A after storage operation?

A. A C W
B. (not C and A) or (C and B)
C. (not C and W) or (C and A)
D. (W or A) and not C
E. not C and W

The essence of storage

Have two memory cells: A, address 0; B, address 1.
Have a control bit, C (specifies address 0 or 1).
Have new bit to write W.
What formula in terms of these values describes new value at B after storage operation?

A. B C W
B. (not C and B) or (C and A)
C. (not C and B) or (C and W)
D. (W or B) and C
E. B and W
Keeping track of instructions

Store instructions in new memory
Keep a “program counter” of current instruction
In each step:
•Retrieve current instruction
•Execute current instruction
•Increment program counter

Visualization

PC → Program
Add 1 → Retrieve
Retrieve → Instruction
**Summary: von Neumann Architecture**

A computer is just a big state machine
- Input: registers, memory, input devices
- Output: new values for registers, memory, output devices
- PC: address of next step to execute

**Power of CPU**

If we build one machine that takes
- truth values for variables
- sequence of simple steps
- and carries out those steps
This machine can compute any function!
We can *program* the machine to do what we want!
Actual CPUs – some differences

Many accumulators – called “registers”
  • Each stores “word” (many bits)
    Nowadays often 32 or 64 bits

Large main memory
  • Use registers to store addresses
  • Say where data should come from, go

Actual CPUs – some differences

Arithmetic operations as well as logic
  • Add, subtract, multiply, divide, compare

Control operations
  • Jump forward or backward to new instruction
  • Conditional on results of operations
Programming

Programming languages:
• a way of writing a kind of script
• rules for the computer to interpret the script as instructions.

Many different ones
• all do pretty much the same things
• make it easier to say some things than others.

Scratch

Developed by the “Lifelong Kindergarten” group at MIT.

Allows users to make media-rich programs by clicking together blocks.

Share your creations a la YouTube.
Scratch programming window

Scratch concepts

Program describes an interactive multimedia performance.

Performance takes place in window called “stage”

Actors in the performance are called “sprites”
Sprites

Basic “objects” of scratch programs
• A lot like a state machine

What’s a sprite:
• Information: name, location (x,y), direction, visibility, “costume” (appearance), tempo…
• Action: move, draw with pen, make sound, change costume, send message, detect event

How a scratch program runs

Send a “start” message to all sprites

Each runs corresponding script
• Executes actions in sequence

Fancy features of scripts
• Messages to invoke other performance elements
• Interesting control structure
Graphical language for steps – motion