

# COMS12200

## Introduction to Computer Architecture

*Simon Hollis (simon@cs.bris.ac.uk)*

# Register machines

- Register machines use multiple storage registers to store sets of data (= variables) at the same time.
- Instructions can typically access multiple registers at once.
- Results are also committed to registers.

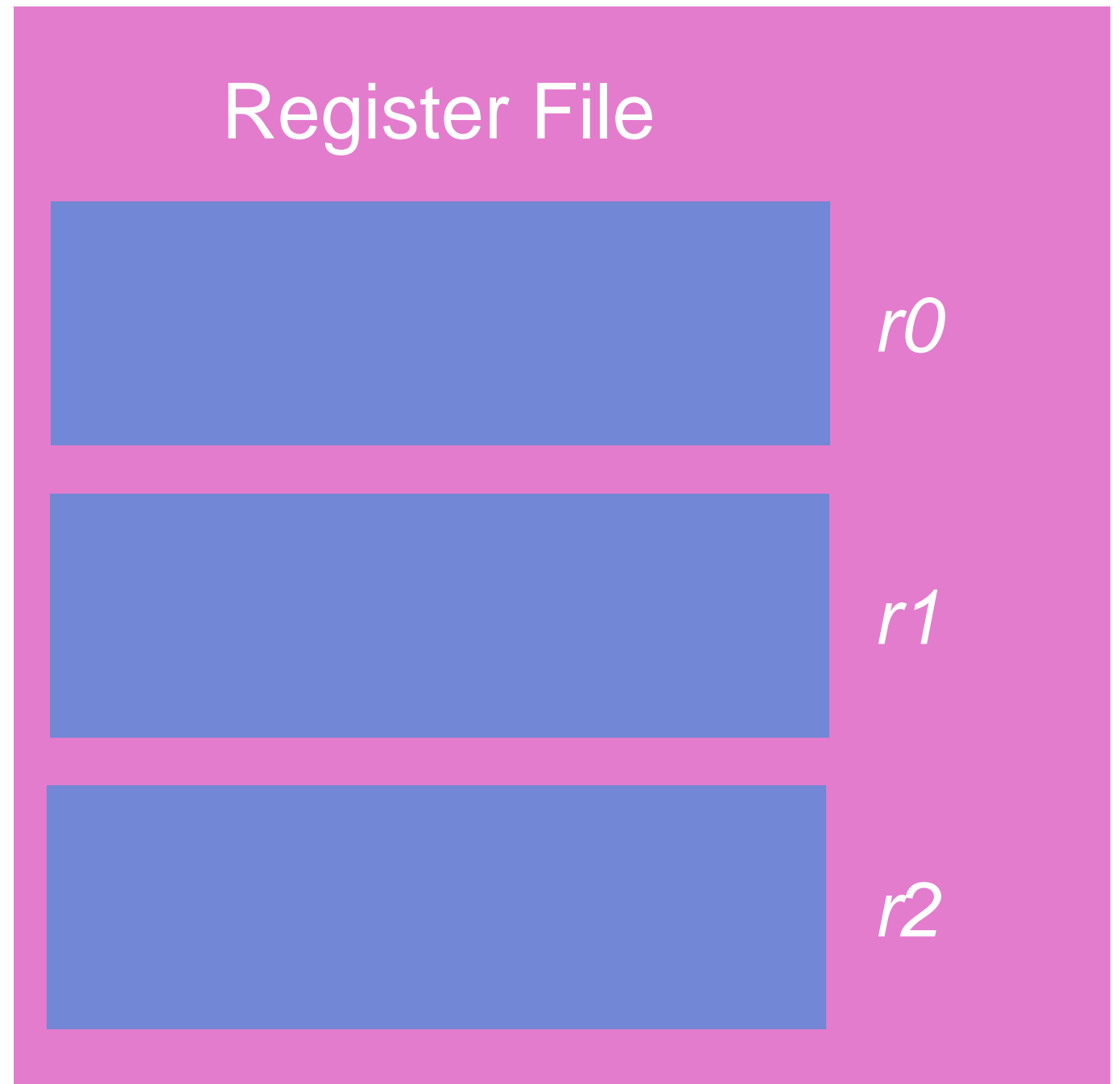
# Register machines

“4 + 2” →

MOVE (r0, 4)

MOVE (r1, 2)

ADD (r0, r0, r1)



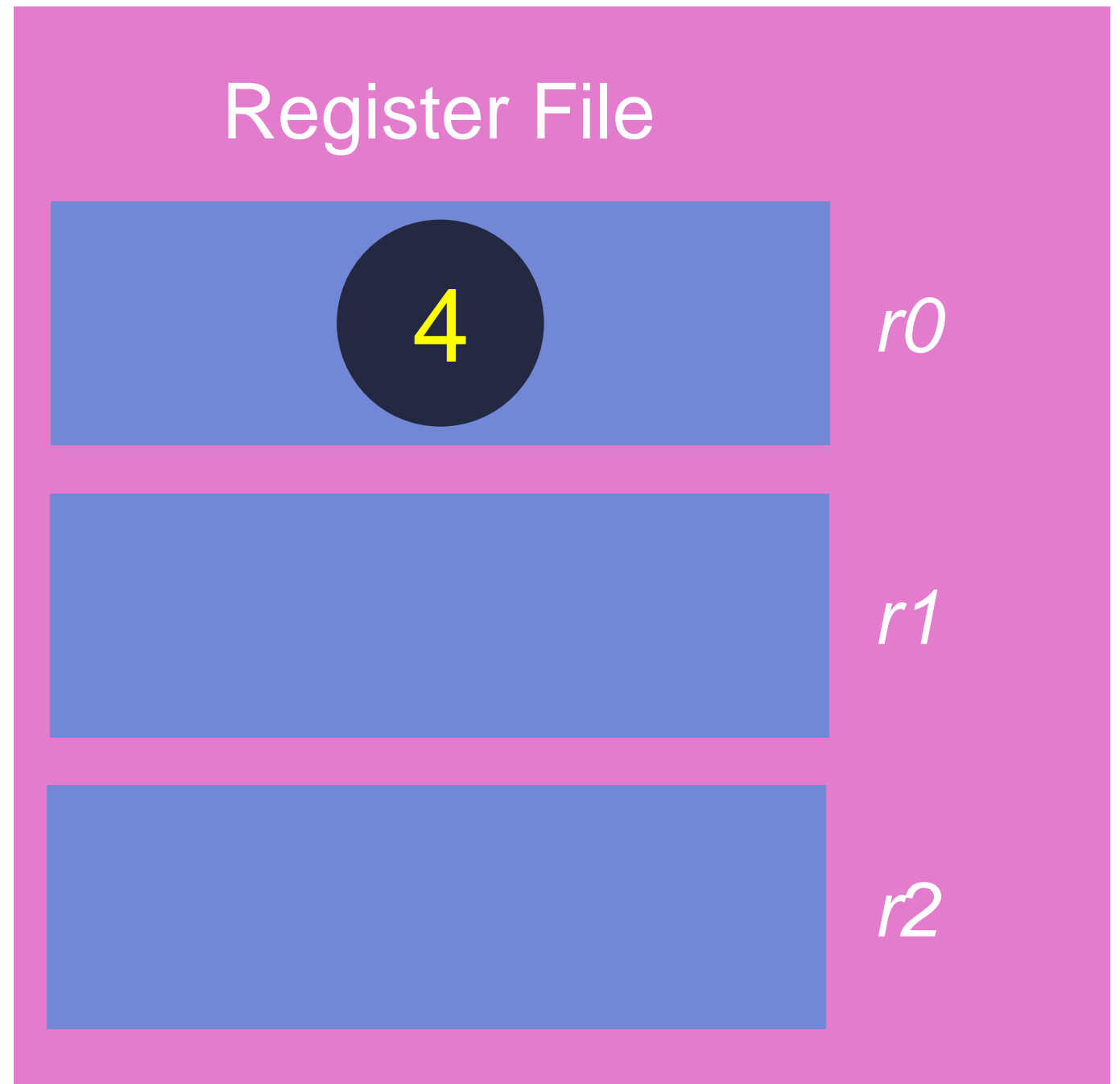
# Register machines

“4 + 2” →

**MOVE (r0, 4)**

MOVE (r1, 2)

ADD (r0, r0, r1)



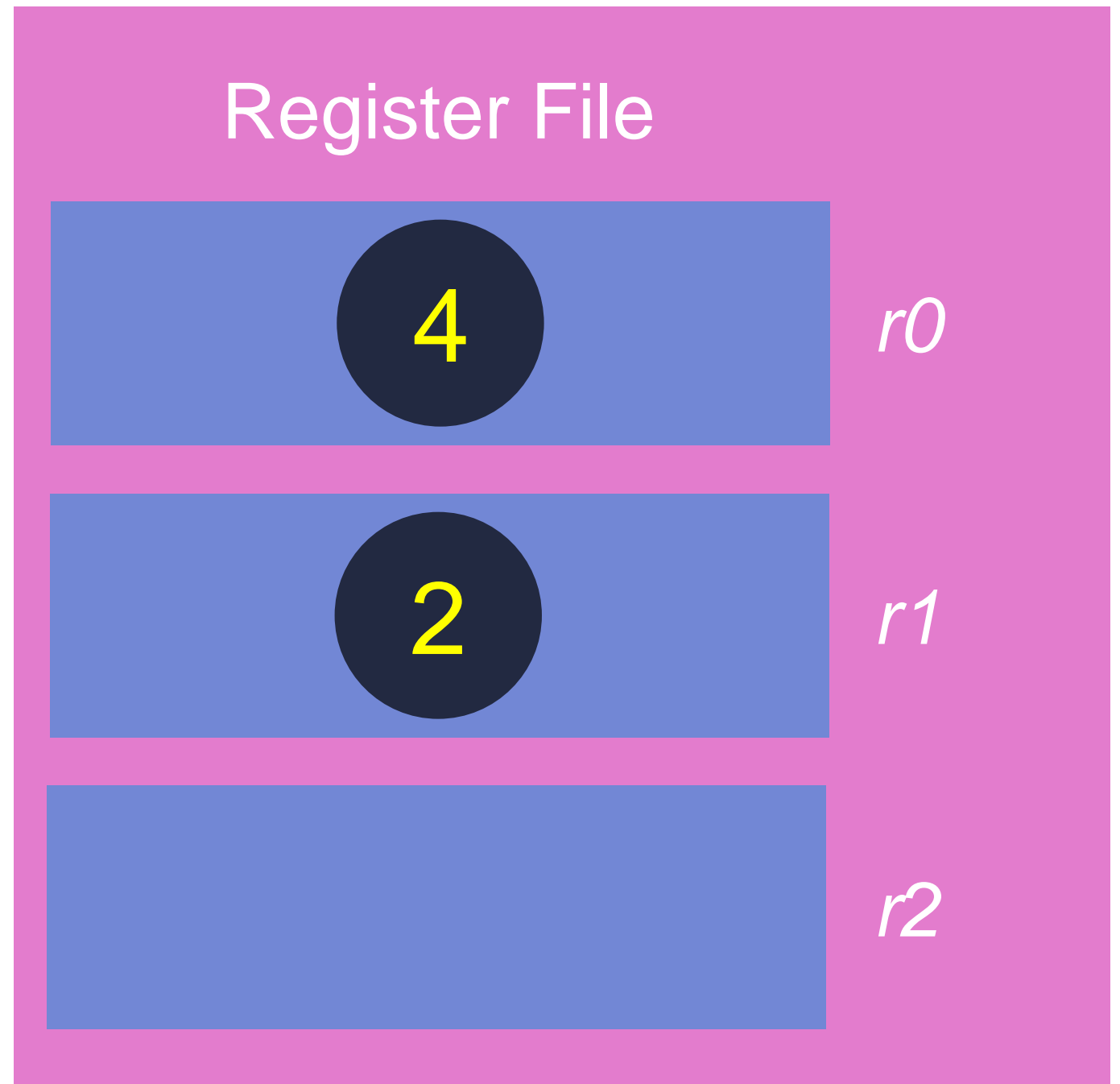
# Register machines

“4 + 2” →

MOVE (r0, 4)

**MOVE (r1, 2)**

ADD (r0, r0, r1)



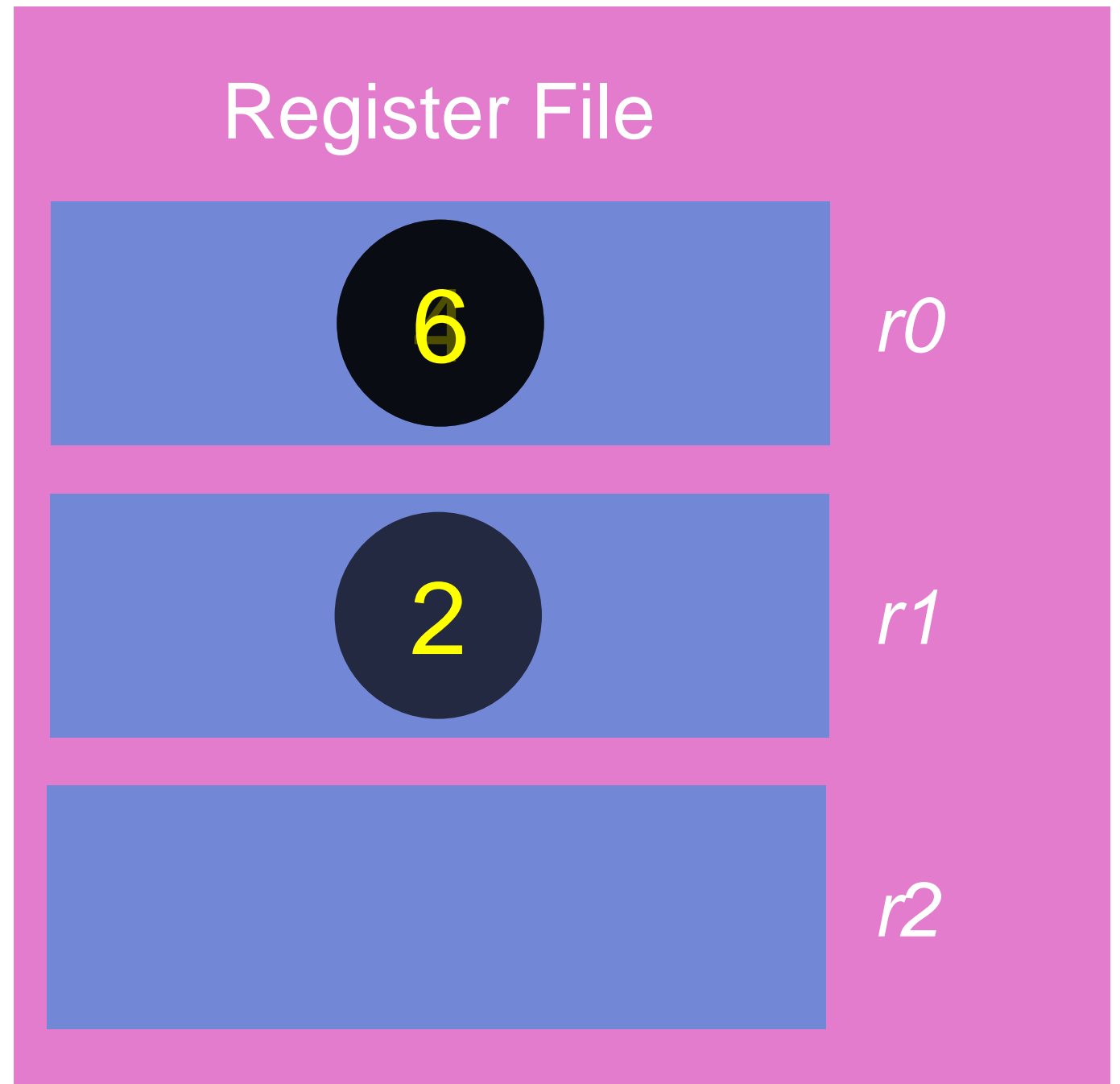
# Register machines

“4 + 2” →

MOVE (r0, 4)

MOVE (r1, 2)

**ADD (r0, r0, r1)**



# Register machine uses

- The majority of modern architectures use the register machine paradigm, due to its flexibility, high performance.
- **Compilers** have become very sophisticated to use registers optimally.
- Intel's **x86**, **ARM** and **MIPS** are all register machines.

# Register sub-flavours

- Depending on the instruction set, one or multiple registers can be simultaneously accessed.
- Some machines only access one, plus memory “*register-memory*” machines
  - e.g. x86 (not a perfect description)
- Many give more flexibility and use multiple registers: “*register-register*” machines
  - e.g. ARM, MIPS