Imperial College
London

Project Brief

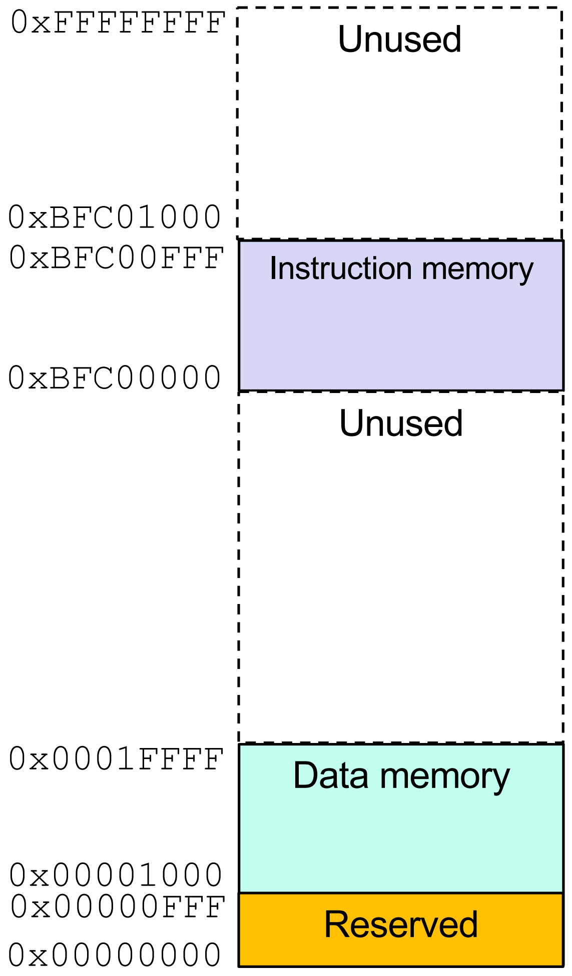
Peter Cheung
Imperial College London

URL: www.ee.imperial.ac.uk/pcheung/teaching/EE2_CAS/
E-mail: p.cheung@imperial.ac.uk

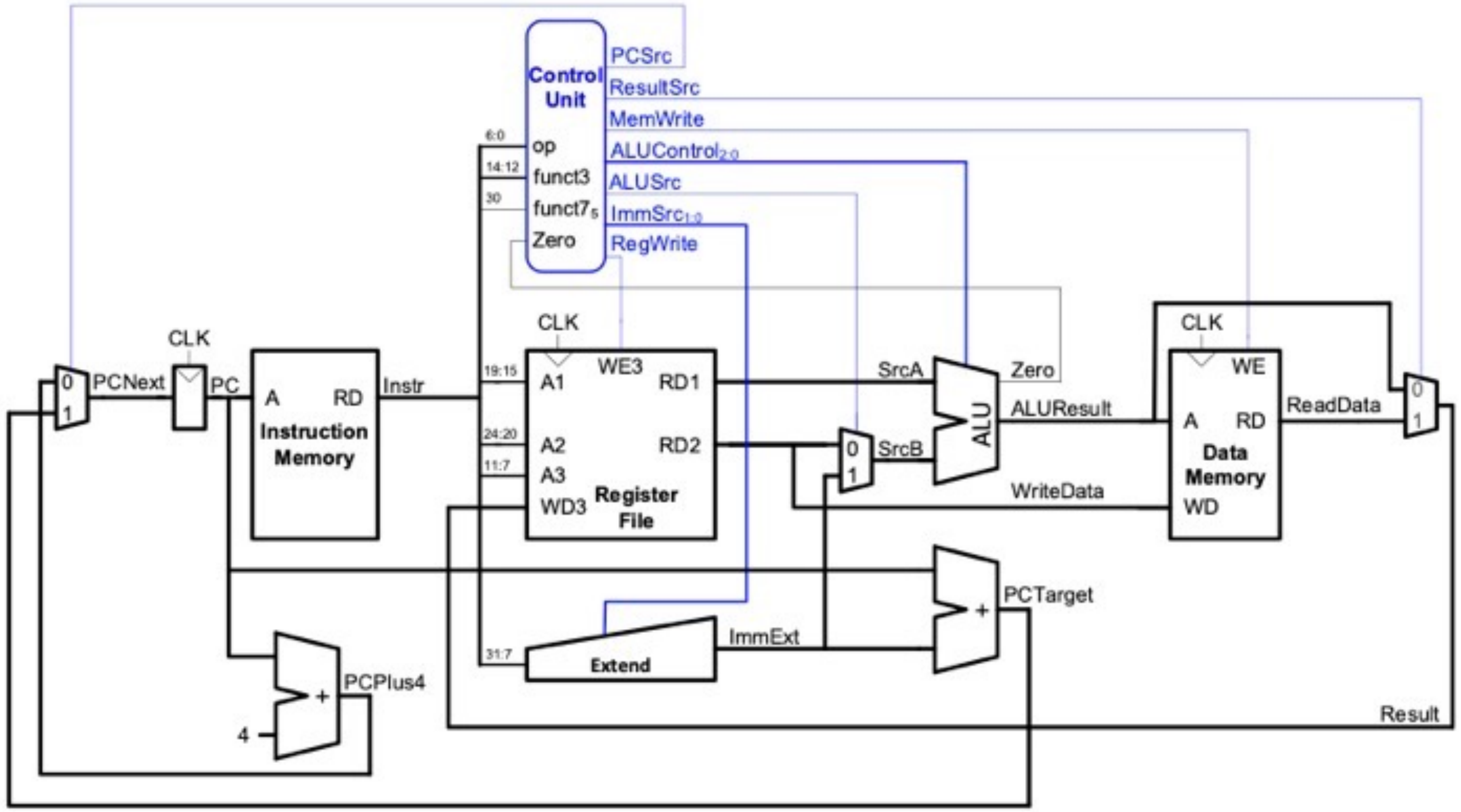
Project Coursework

- **Basic**
  - F1 starting light algorithm from Lab 3 implemented in RISC-V assembly code
  - Design of a single-cycle RISC-V RV32I processor that runs your code + a reference program which compute the probability distribution function of some signal waveforms
- **Stretched goal 1**
  - A pipelined version of the single-cycle RISC-V with hazard handling
- **Stretched goal 2**
  - Add set-associative data cache to the pipelined processor
- **Even more stretched goal – use your imagination**

Memory Map for your Project

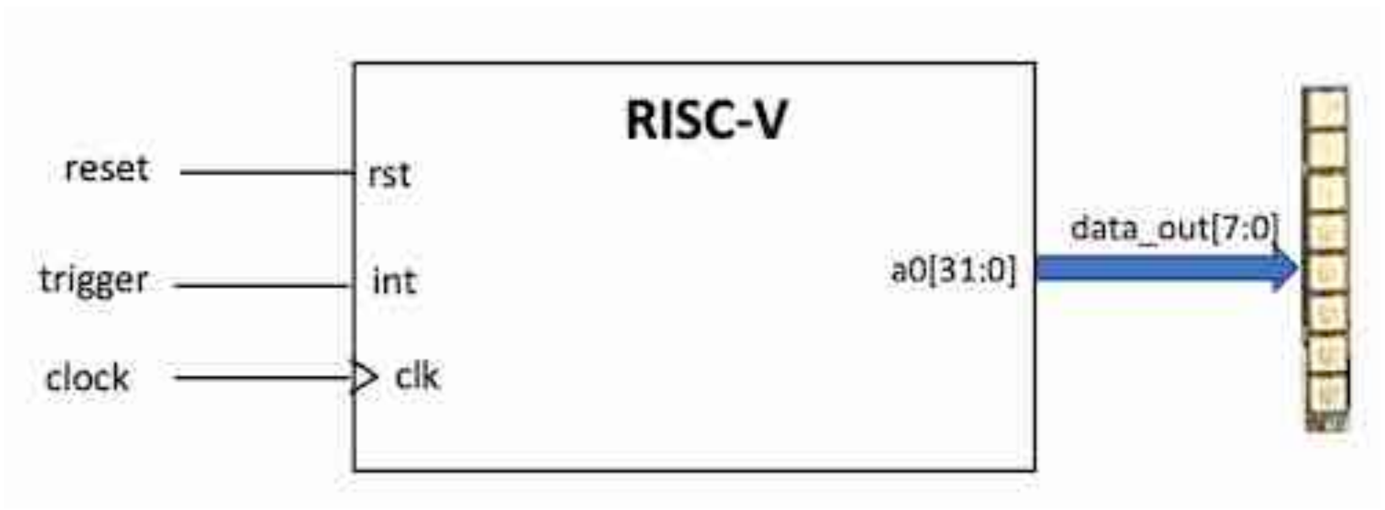


Single-cycle RISC-V RV32I

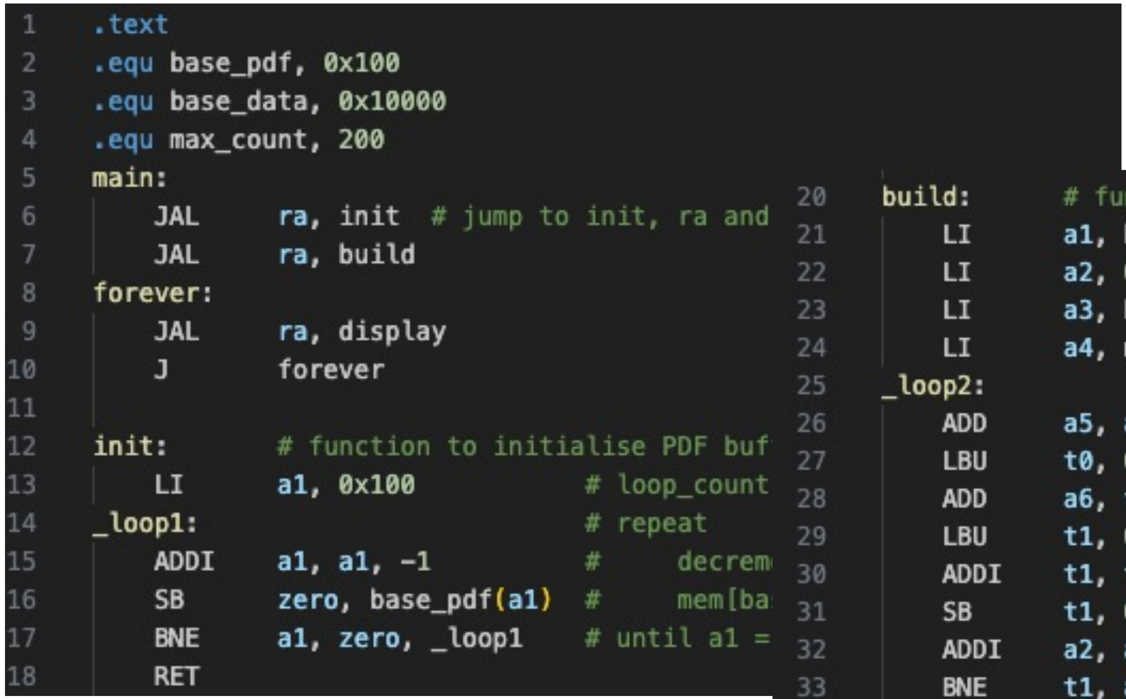


Based on: "*Digital Design and Computer Architecture (RISC-V Edition)*"
by Sarah Harris and David Harris (H&H),

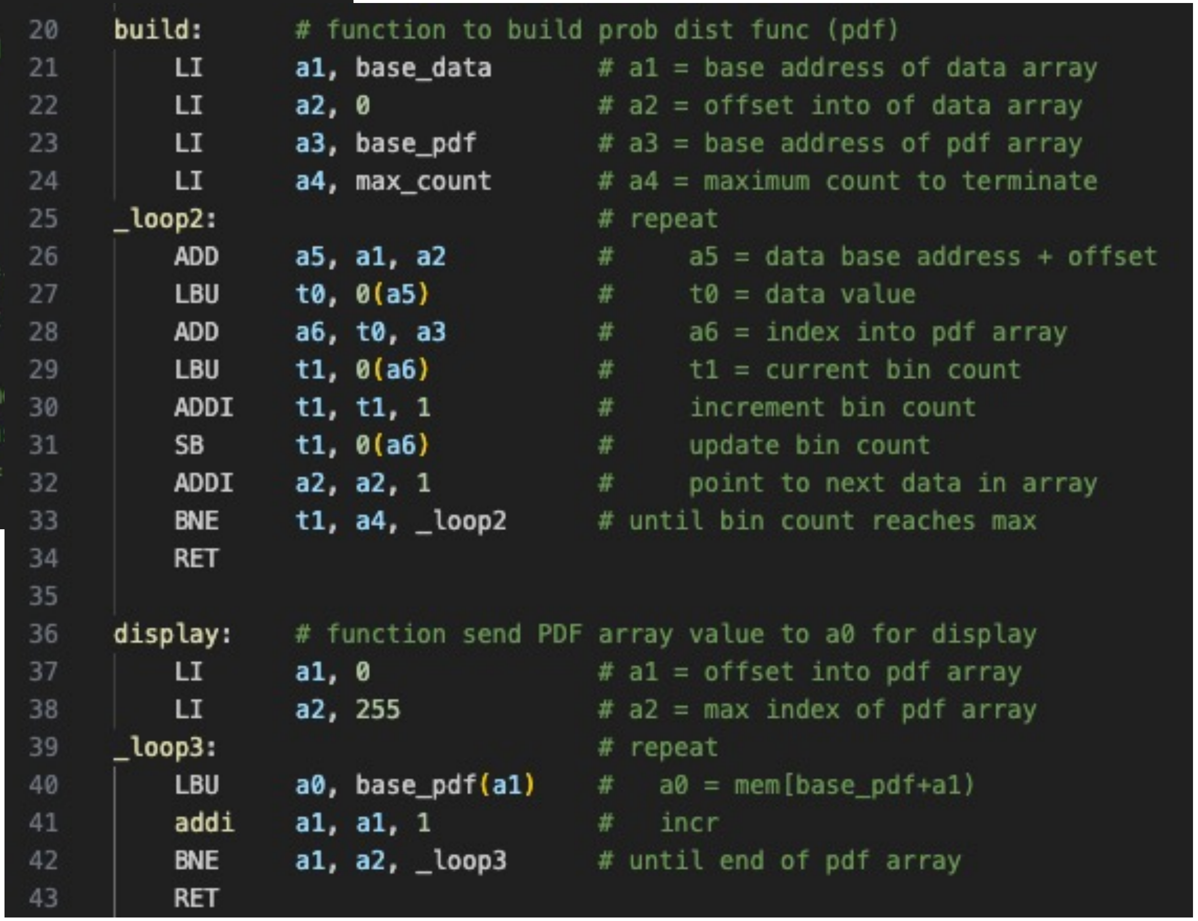
Verification 1: F1 on RISC-V



Verification 2: Reference Program – pdf.s

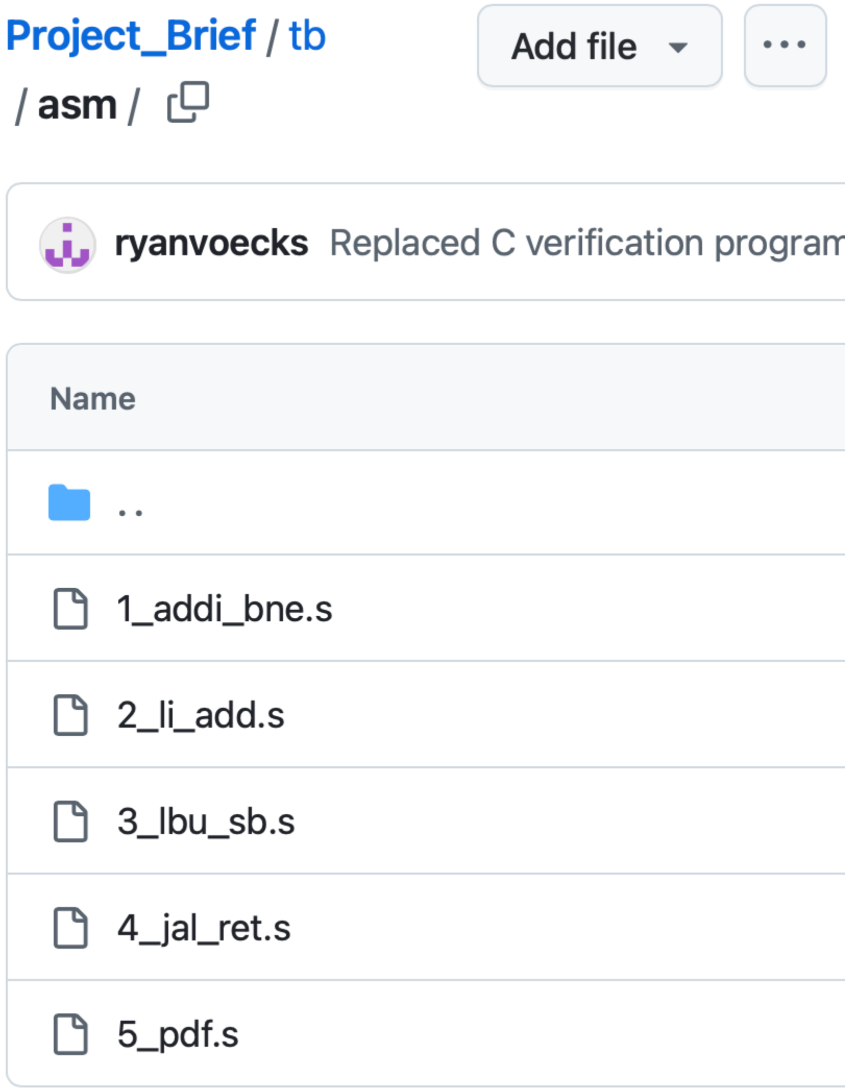


```
.text
.equ base_pdf, 0x100
.equ base_data, 0x10000
.equ max_count, 200
main:
    JAL     ra, init  # jump to init, ra and
    JAL     ra, build
forever:
    JAL     ra, display
    J       forever

init:       # function to initialise PDF buf
    LI      a1, 0x100           # loop_count
_loop1:                         # repeat
    ADDI    a1, a1, -1          #     decrem
    SB      zero, base_pdf(a1)  #     mem[ba
    BNE     a1, zero, _loop1    # until a1 =
    RET
```



```
build:      # function to build prob dist func (pdf)
    LI      a1, base_data       # a1 = base address of data array
    LI      a2, 0               # a2 = offset into of data array
    LI      a3, base_pdf        # a3 = base address of pdf array
    LI      a4, max_count       # a4 = maximum count to terminate
_loop2:                         # repeat
    ADD     a5, a1, a2          #     a5 = data base address + offset
    LBU     t0, 0(a5)           #     t0 = data value
    ADD     a6, t0, a3          #     a6 = index into pdf array
    LBU     t1, 0(a6)           #     t1 = current bin count
    ADDI    t1, t1, 1           #     increment bin count
    SB      t1, 0(a6)           #     update bin count
    ADDI    a2, a2, 1           #     point to next data in array
    BNE     t1, a4, _loop2      # until bin count reaches max
    RET

display:    # function send PDF array value to a0 for display
    LI      a1, 0               # a1 = offset into pdf array
    LI      a2, 255             # a2 = max index of pdf array
_loop3:                         # repeat
    LBU     a0, base_pdf(a1)    #   a0 = mem[base_pdf+a1)
    addi    a1, a1, 1           #   incr
    BNE     a1, a2, _loop3      # until end of pdf array
    RET
```

Verification 3: Test programs


Project_Brief / tb / asm /

Add file

ryanvoecks Replaced C verification progran

Name

..

1_addi_bne.s

2_li_add.s

3_lbu_sb.s

4_jal_ret.s

5_pdf.s

Deliverables

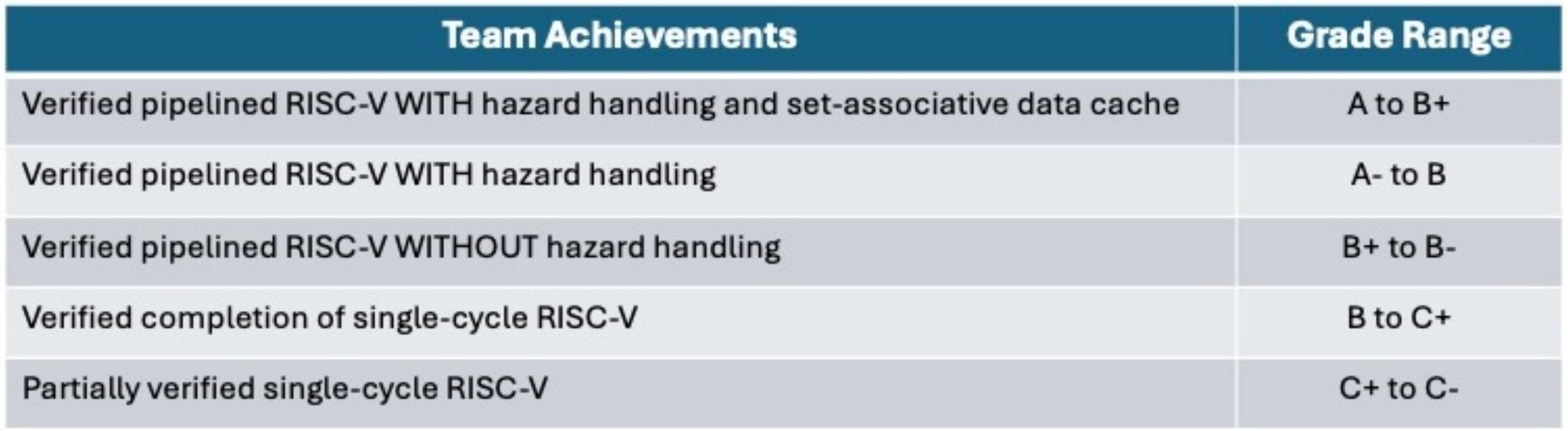
- All deliverables due at 23.59, Friday 13 December 2024
- All deliverables via your team’s GitHub repo which your team has set up

Deliverables must include:

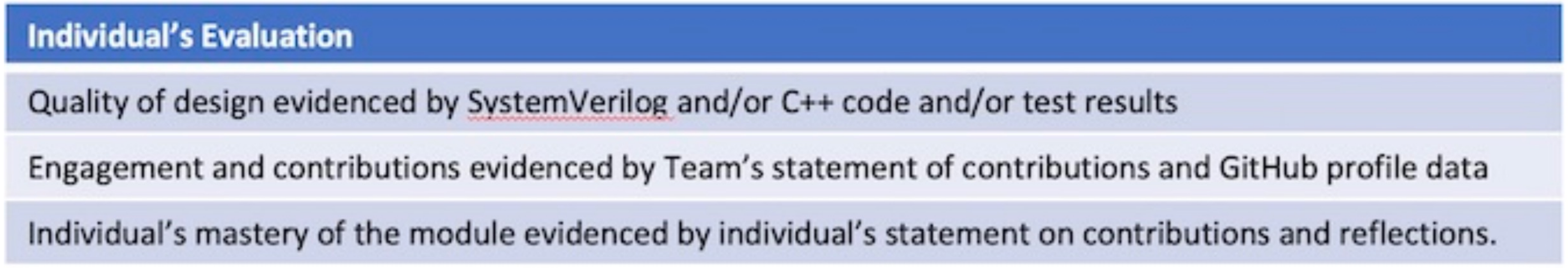
1. **README.md – joint statement** on what team has achieved and agreed statement of each person’s contributions.
2. Individual **personal statement** on contributions, what you have learned, mistakes made, how you might do differently – reflections.
3. A **rtl** or source folder of your team’s design.
4. A **test folder** with evidence that your processor works on F1 program, the reference program “**pdf.s**”, and other test programs in assembly language.

You must also include a **makefile** or **shell** script to build all your designs and run the testbench that allows me to repeat what you have done.

Assessment criteria



Team Achievements	Grade Range
Verified pipelined RISC-V WITH hazard handling and set-associative data cache	A to B+
Verified pipelined RISC-V WITH hazard handling	A- to B
Verified pipelined RISC-V WITHOUT hazard handling	B+ to B-
Verified completion of single-cycle RISC-V	B to C+
Partially verified single-cycle RISC-V	C+ to C-



Individual's Evaluation
Quality of design evidenced by SystemVerilog and/or C++ code and/or test results
Engagement and contributions evidenced by Team's statement of contributions and GitHub profile data
Individual's mastery of the module evidenced by individual's statement on contributions and reflections.