CS152/252A
Fall 2026

Architectures, Microcoding

Discussion 1

1 Architecture vs Microarchitecture

True or false: The following is architecturally visible (exposed by the architecture)?

(a) Register file entries in a classical RISC pipeline

(b) The stack in a stack architecture

(c) Pipeline registers

(d) Branch-delay / load-delay slots

(e) NOPs

(f) Pipeline bubbles

(g) Condition codes, status flags

(h) Memory address width

(i) Instruction/data caches

2 Microcoded vs Pipelined

How does a microcoded machine differ from a classic RISC pipeline?

Why is a simpler microarchitecture generally possible with microcoding?

3 Microprogramming

Implement a conditional memory-to-memory move instruction in microcode for the single-bus RISC-V machine described in Handout 1 The instruction has the following format:

```
CMOVM (rd), (rs1), rs2
```

CMOVM performs the following operation: If the value in rs2 is true (non-zero), then the memory word loaded from the address in rs1 is stored to the address in rd.

```
if R[rs2] != 0
  M[R[rd]] := M[R[rs1]]
```

Fill in the following table with the microinstructions and control signals. Optimize your microprogram to minimize the number of cycles and to set entries to don't-cares (*) wherever possible.

State	Pseudocode	Ir Ld	Reg Sel	Reg Wr	Reg En	A Ld	B Ld	ALU Op	ALU En	MA Ld	Mem Wr	Mem En	Imm Sel	Imm En	µBr	Next State
FETCH0	MA := PC A := PC	*	PC	0	1	1	*	*	0	1	0	0	*	0	N	*
	IR := Mem	1	*	0	0	0	*	*	0	0	0	1	*	0	S	*
	PC := A+4	0	PC	1	0	0	*	INC_A_4	1	*	0	0	*	0	D	*
…																
NOP0	µBr to FETCH0	*	*	0	0	*	*	*	0	*	0	0	*	0	J	FETCH0
CMOVM0:																
…																