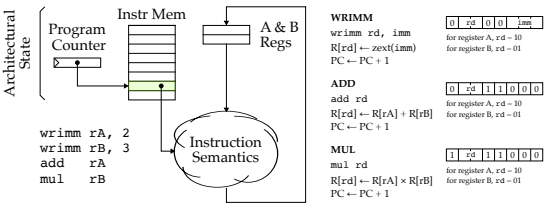
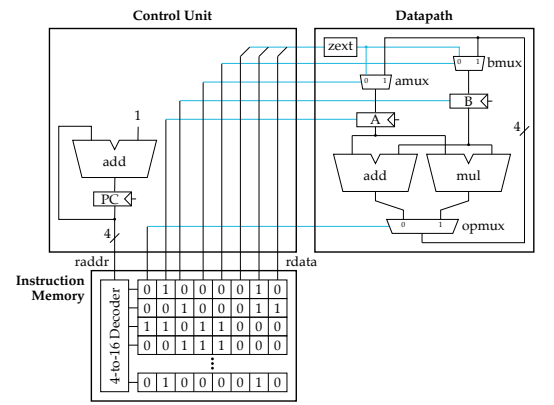
3. From Architecture to Microarchitecture

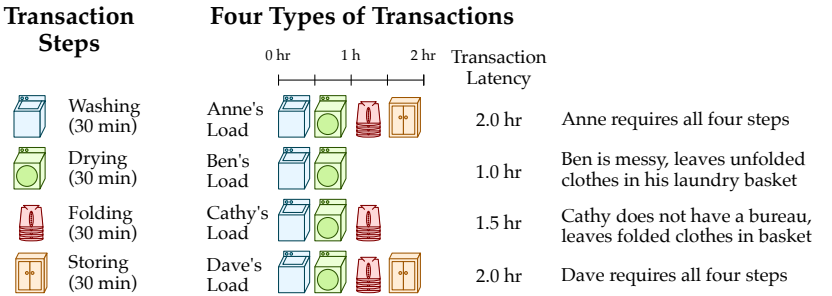
- Processor
  - Instructions are "transactions" that execute on a processor
  - Architecture: defines the hardware/software interface
  - Microarchitecture: how hardware executes sequence of instructions



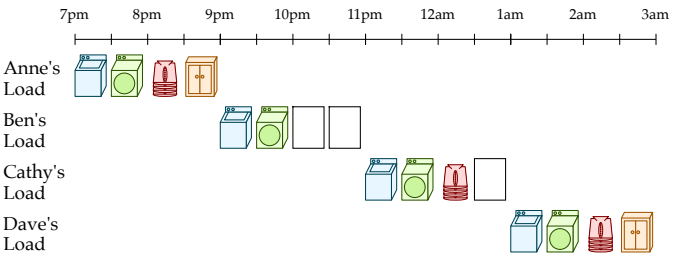


- Laundry
  - Cleaning a load of laundry is a "transaction"
  - Architecture: high-level specification, dirty clothes in, clean clothes out
  - Microarchitecture: how laundry room actually processes multiple loads

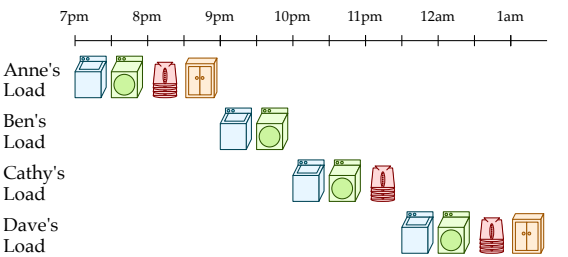
3.1. Processor Microarchitectural Design Patterns



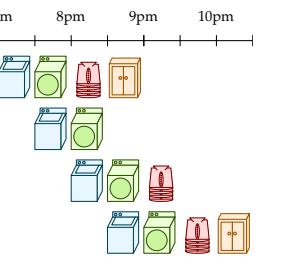
Fixed Time Slot Laundry (Single-Cycle Processors)



Variable Time Slot Laundry (Multi-Cycle Processors)

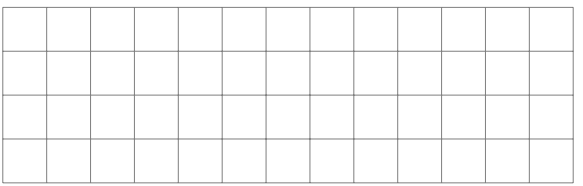


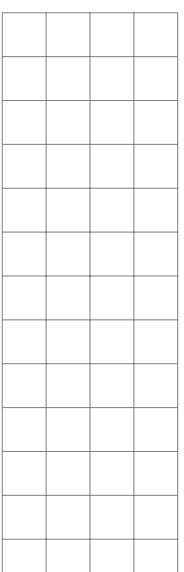
Pipelined Laundry



3.2. Transaction Diagrams

 **W**: Washing  **D**: Drying  **F**: Folding  **S**: Storing





Key Concepts

- Transaction latency is the time to complete a single transaction
- Execution time or total latency is the time to complete a sequence of transactions
- Throughput is the number of transactions executed per unit time

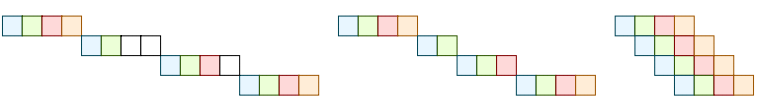
3.3. Analyzing Processor Performance

$$\frac{\text{Time}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Avg Cycles}}{\text{Instruction}} \times \frac{\text{Time}}{\text{Cycle}}$$

- Instructions / program depends on source code, compiler, ISA
- Avg cycles / instruction (CPI) depends on ISA, microarchitecture
- Time / cycle depends upon microarchitecture and implementation

Using our first-order equation for processor performance and a functional-level model, the execution time is just the number of dynamic instructions.

Microarchitecture	CPI	Cycle Time
Single-Cycle Processor	1	long
Multi-Cycle Processor	>1	short
Pipelined Processor	≈1	short



Students often confuse "Cycle Time" with the execution time of a sequence of transactions measured in cycles. "Cycle Time" is the clock period or the inverse of the clock frequency.

Estimating dynamic instruction count

Estimate the dynamic instruction count for the vector-vector add assembly program assuming `n` is 64?

```
1  loop:
2   lw    x5, 0(x1)
3   lw    x6, 0(x2)
4   add   x7, x5, x6
5   sw    x7, 0(x3)
6   addi  x1, x1, 4
7   addi  x2, x2, 4
8   addi  x3, x3, 4
9   addi  x4, x4, -1
10  bne   x4, x0, loop
```

Estimate the dynamic instruction count for the find assembly program assuming `n` is 64 and only the first element matches the given value.

```
1   addi  x5, x0, 0
2
3  loop:
4   lw    x4, 0(x1)
5   bne   x4, x3, foo
6   addi  x5, x0, 1
7
8  foo:
9   addi  x1, x1, 4
10  addi  x2, x2, -1
11  bne   x2, x0, loop
```