

## Pipelining

- Bathroom example
  - 3 stations, 1 minute each, 5 people, takes 7 minutes
  - with 6 stations at 30 seconds each, it takes 2.5 minutes to fill + 5 minutes to empty = 7.5 minutes
  - The process will be no faster than the slowest step
- Instruction cycle
  - Fetch instruction
  - Decode
  - Fetch operands
  - Execute
  - Store or "Write" operands
  - Instruction fetch, operand fetch, and operand store all access the bus
- Math
  - without pipelining = stages per instruction \* # of instructions
  - with pipe (prefetch) = stages to fill the pipe + # of instructions
  - As  $n$  goes to infinity (+1 drops out)  $\rightarrow (2 * n) / n = 2$
  - Theoretically, the more instructions can be broken up, the more efficient the pipelining
  - Assumes that FI, FO, and WO can be simultaneous
- Problems
  - Conditional branching can cause refetch of instructions
  - The more stages, the more conditional branching hurts
  - Overhead between stages
- In Class Exercise
  - $1.5 \times 10^6$  instructions
  - 6-stage pipeline
  - 10% chance of a conditional branch
  - 50% chance that a branch will be taken
  - $1.5 \times 10^6 + 5$  = pipelining cycles with no conditional branches
  - $1.5 \times 10^6 + 5 + 4 * (1.5 * 10^6 * 0.05)$ 
    - 4 = stages added due to branch
    - 0.05 = 50% of 10% of instructions
  -