PIPELINING

Arsitektur Komputer

PIPELINING

- Pencapaian performa tinggi komputer merupakan salah satu tujuan utama seorang arsitek komputer.
- Penyediaan teknik *overlap* dan paralelisme dalam komputer merupakan suatu teknik standar untuk meningkatkan performa komputer.
- Teknik lain yang digunakan adalah multiprosesor dalam sebuah sistem komputer.
- Desain unit kontrol menjadi kompleks karena beberapa datapath beroperasi secara simultan.

What is Pipelining?

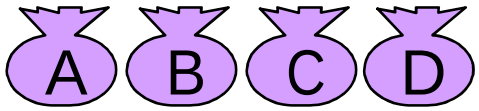
- Pipelining adalah teknik pemecahan satu pekerjaan/ tugas menjadi beberapa sub-tugas, dan mengeksekusi sub-tugas tersebut secara bersamaan/ paralel dalam unit-unit multi hardware atau segmen-segmen

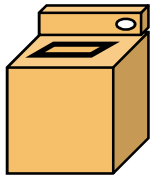
What are the benefits?

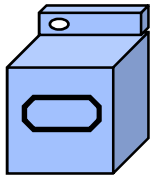
- Tujuan yg ingin dicapai dlm pipeline adalah untuk meningkatkan *throughput*. (the number of instructions complete per unit of time - but it is not reduce the execution time of an individual instruction)
- Waktu yg digunakan untuk eksekusi setiap tugas sama dengan waktu yg digunakan untuk satu eksekusi *non-pipeline.*
- Tetapi karena eksekusi tugas yg berurutan dilakukan secara bersamaan, maka jumlah tugas yg dapat dieksekusi dlm suatu waktu yg disediakan lebih tinggi
- *Hardware pipeline* menyediakan *throughput* yang lebih baik dibandingkan dgn *hardware non-pipeline*.

Pipeline Analogy

- Laundry Example
- Ann, Brian, Cathy, Dave each have one load of clothes to wash, dry, and fold
- Washer takes 30 minutes
- Dryer takes 40 minutes
- “Folder” takes 20 minutes

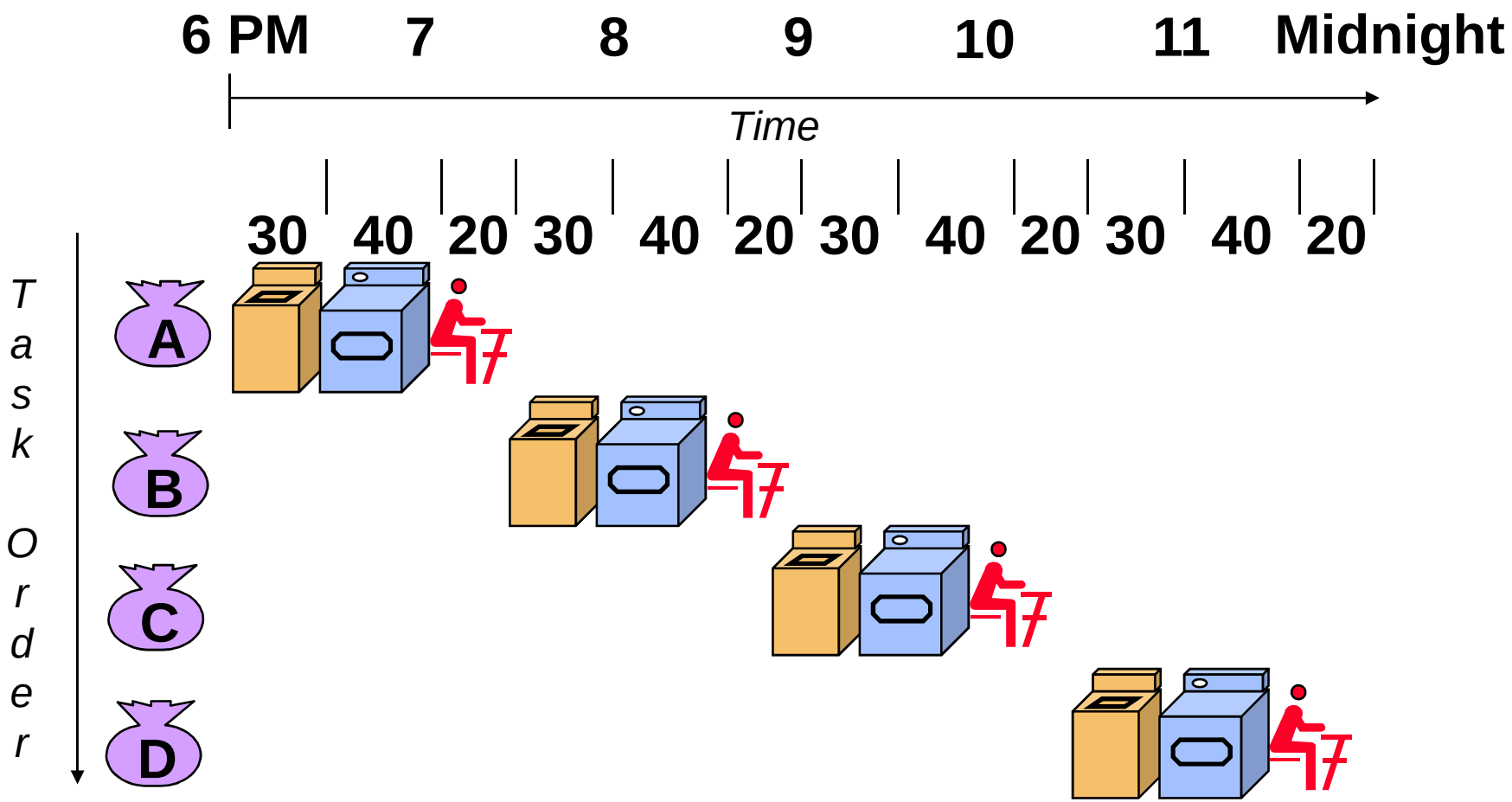






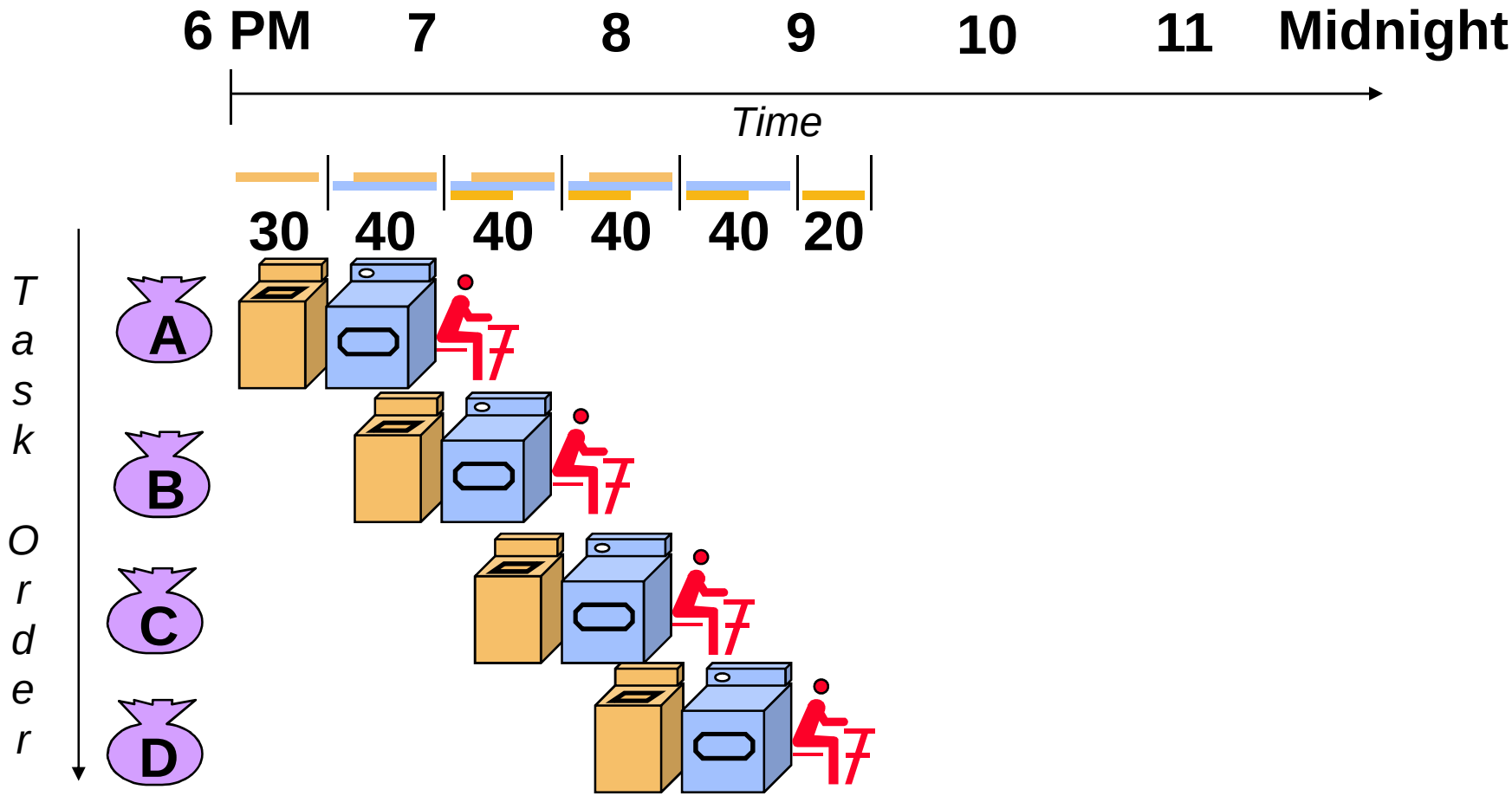


Pipeline Analogy



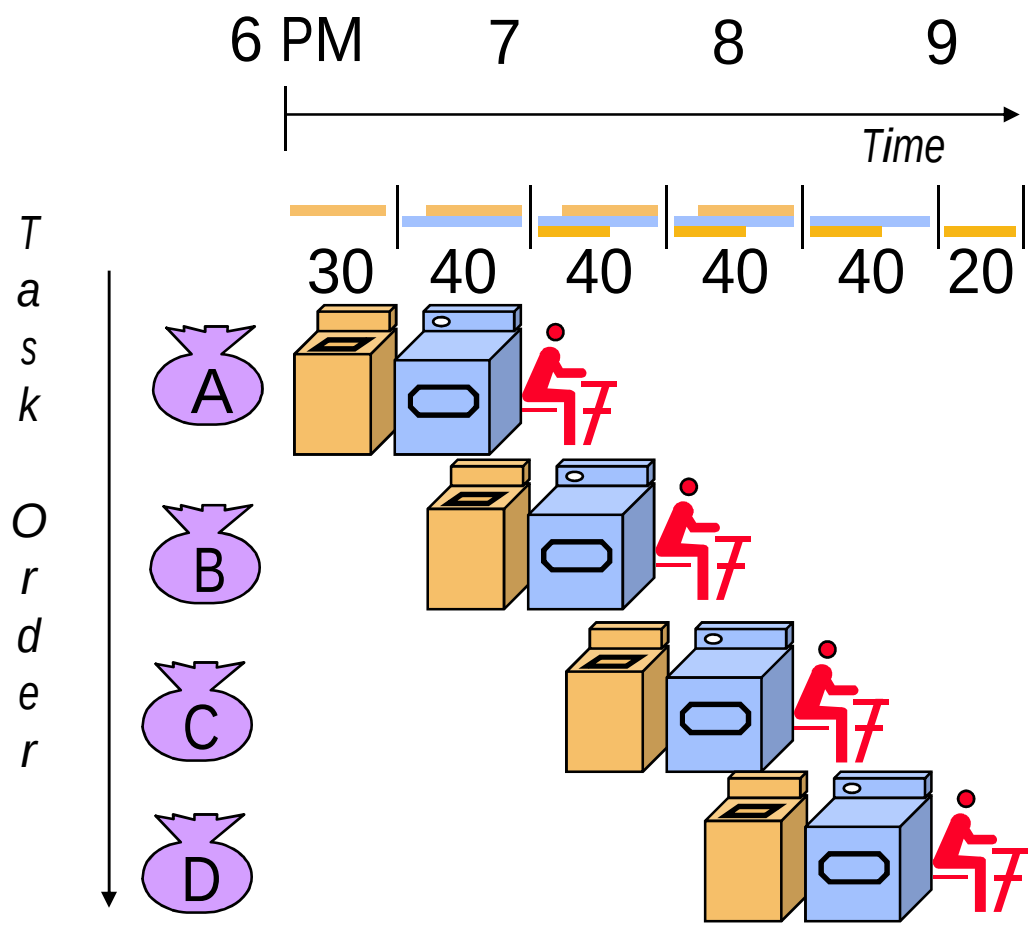
- Sequential laundry takes 6 hours for 4 loads
- If they learned pipelining, how long would laundry take?

Pipeline Analogy



- Pipelined laundry takes 3.5 hours for 4 loads

Pipeline Analogy



- Pipelining tdk membantu latency pekerjaan tunggal, hanya membantu peningkatan throughput untuk beban kerja secara keseluruhan
- Pipeline rate dibatasi oleh tingkat pipeline terendah
- *Multiple tasks* bekerja secara simultan
- Potential speedup = Number pipe stages
- Unbalanced lengths of pipe stages reduces speedup
- Time to "fill" pipeline and time to "drain" it reduces speedup

Fundamental Execution Cycle



baca instruksi dari memori program

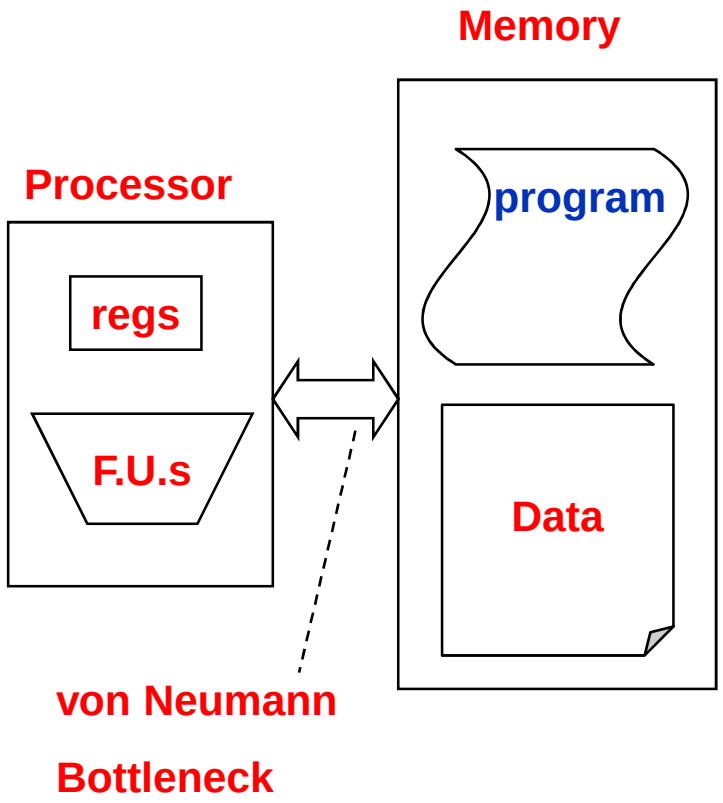
Menetapkan aksi-aksi yg diperlukan dan ukuran instruksi

Menetapkan lokasi dan pengambilan operand (data)

Memproses nilai hasil atau status

menyimpan hasil pada memori

menetapkan instruksi berikutnya

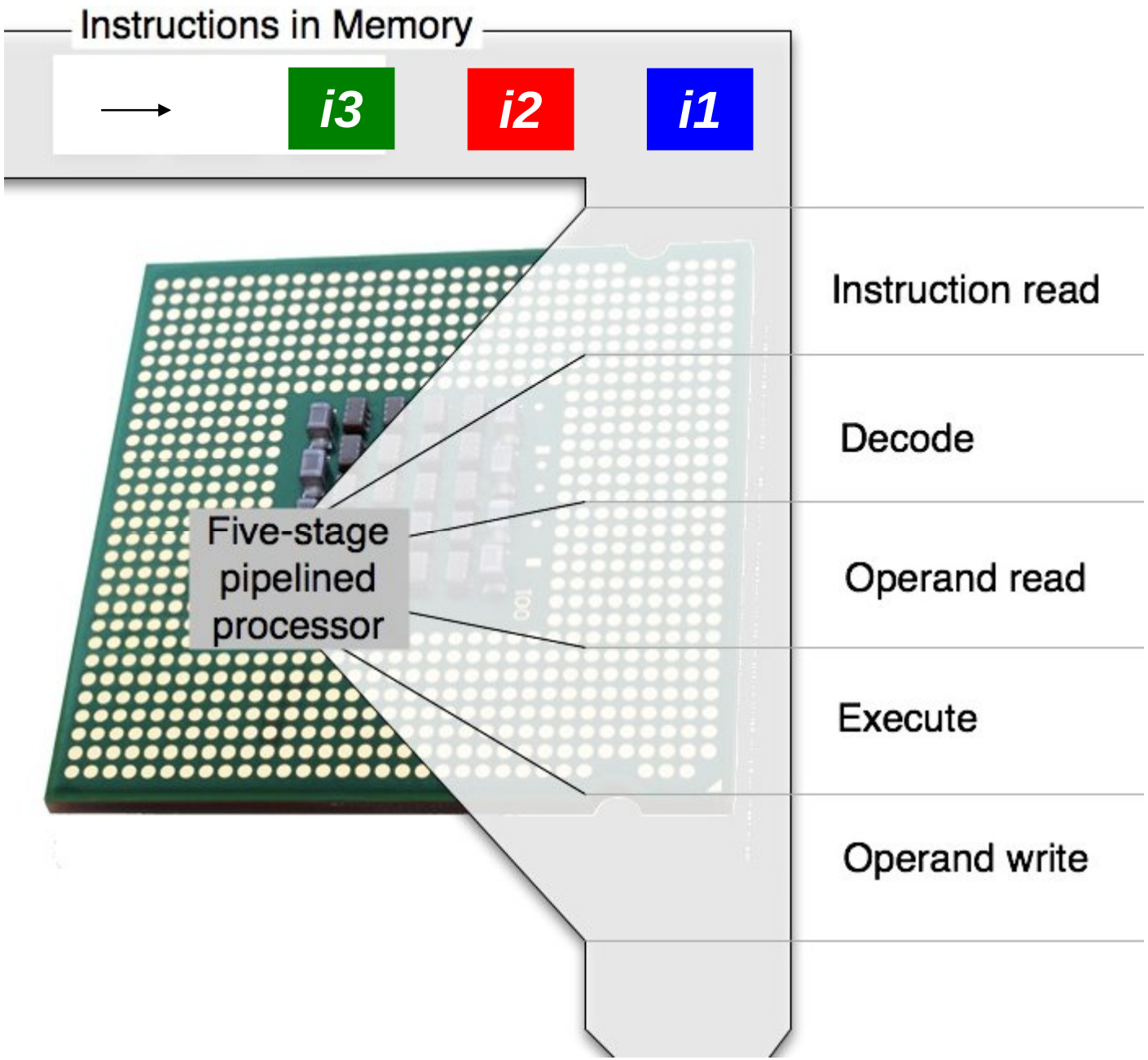


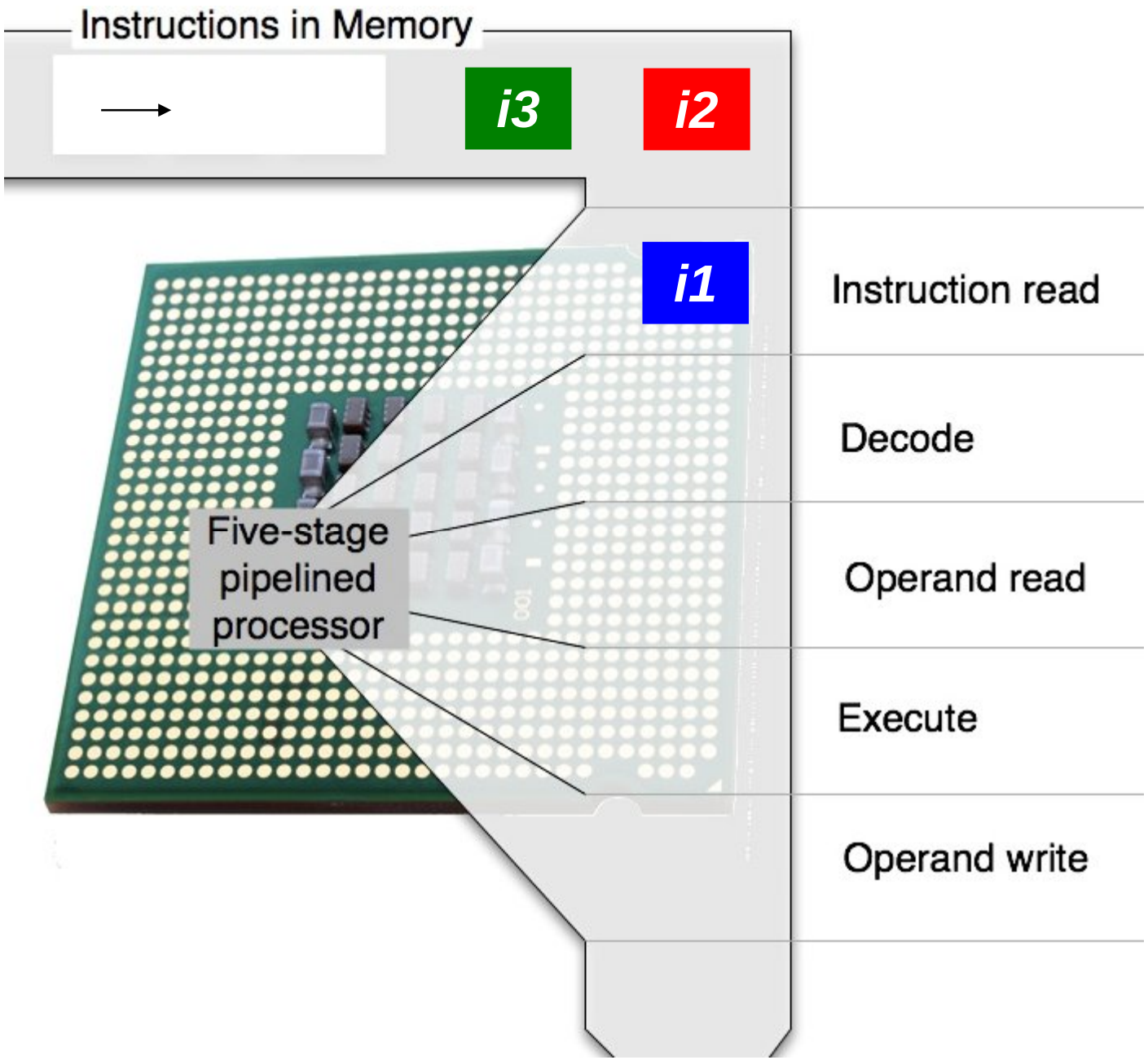
What is pipelining?

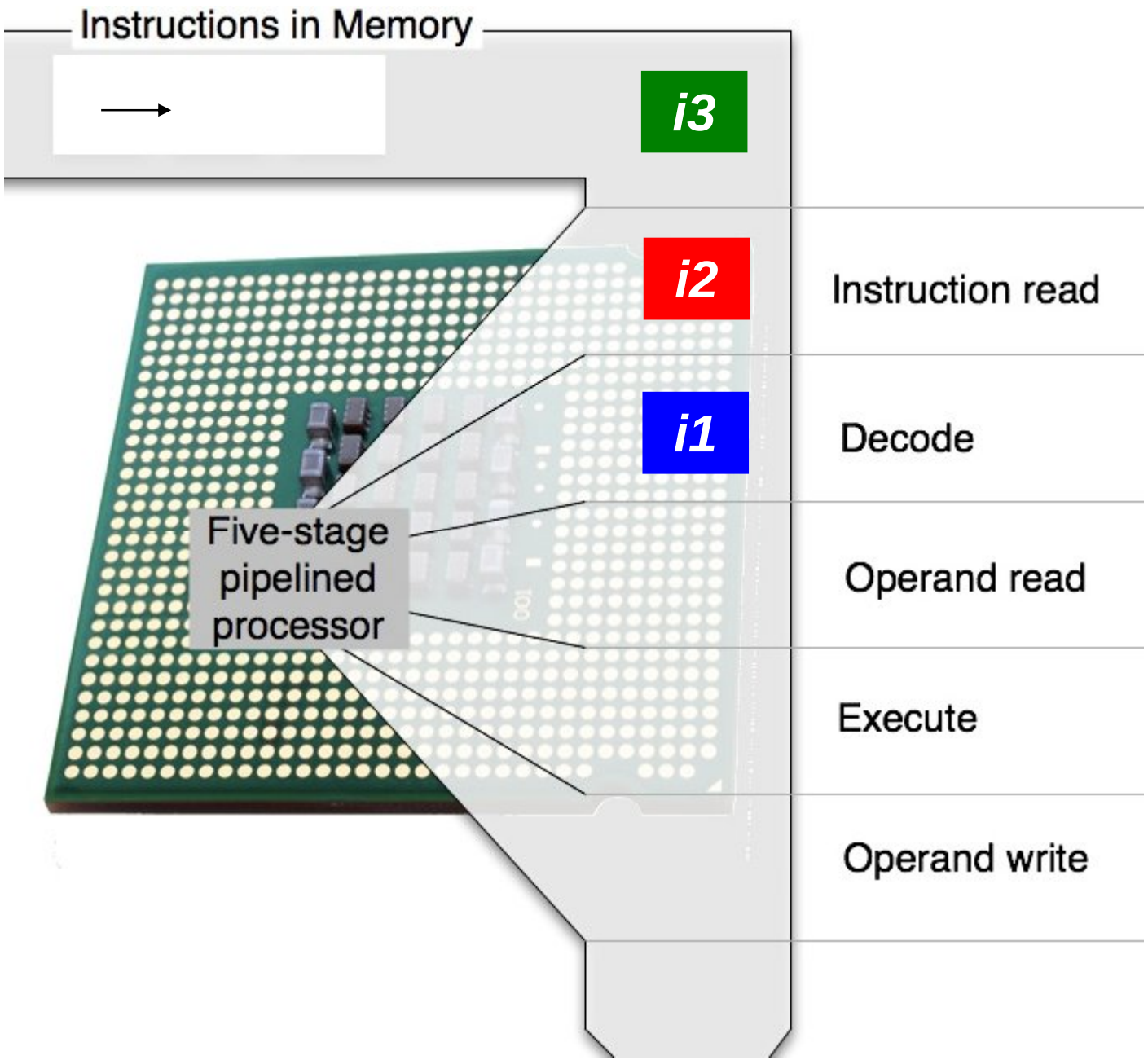
- No more “one instruction at a time” processing
- Processor works simultaneously on multiple instructions
- Each instruction is at a different stage (or is being delayed until its next stage is open)
  - First instruction is fetched from memory
  - First instruction is decoded;
    second instruction is fetched
  - First instruction’s operands are fetched;
    second instruction is decoded;
    third instruction is fetched
  - And on, and on, and on…

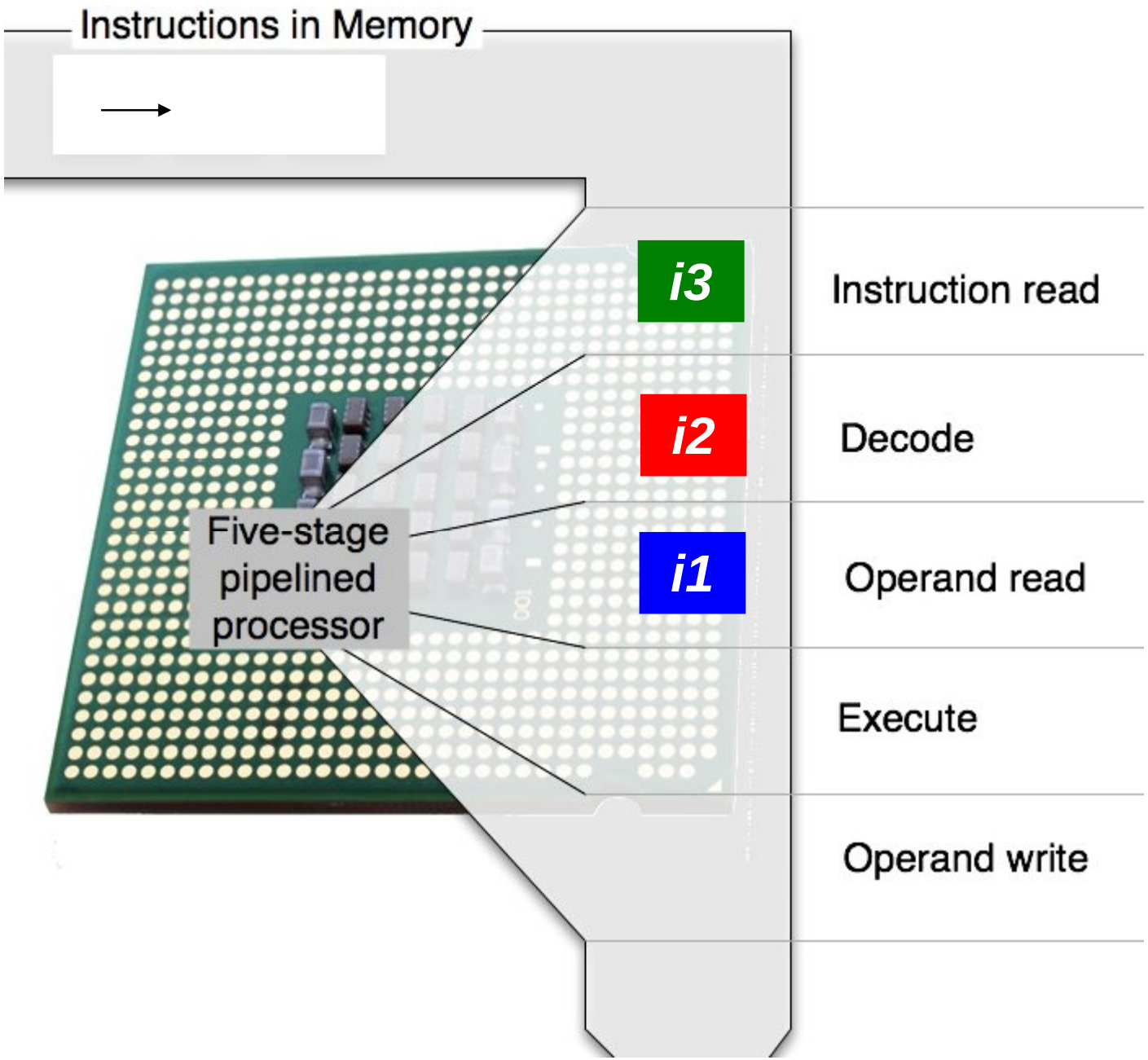
How does pipelining work?

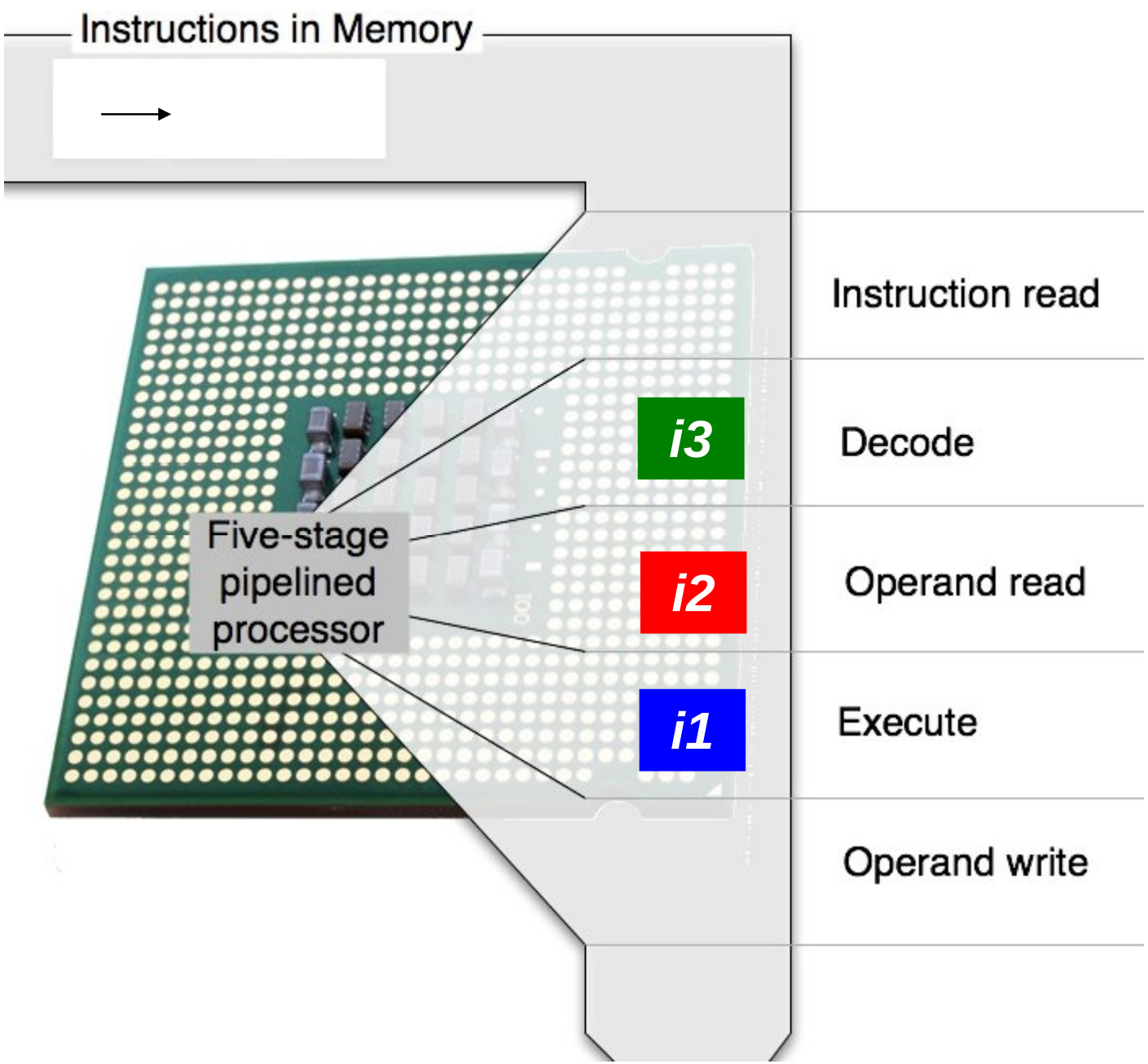
A short(er) animation
of pipelined processor routines

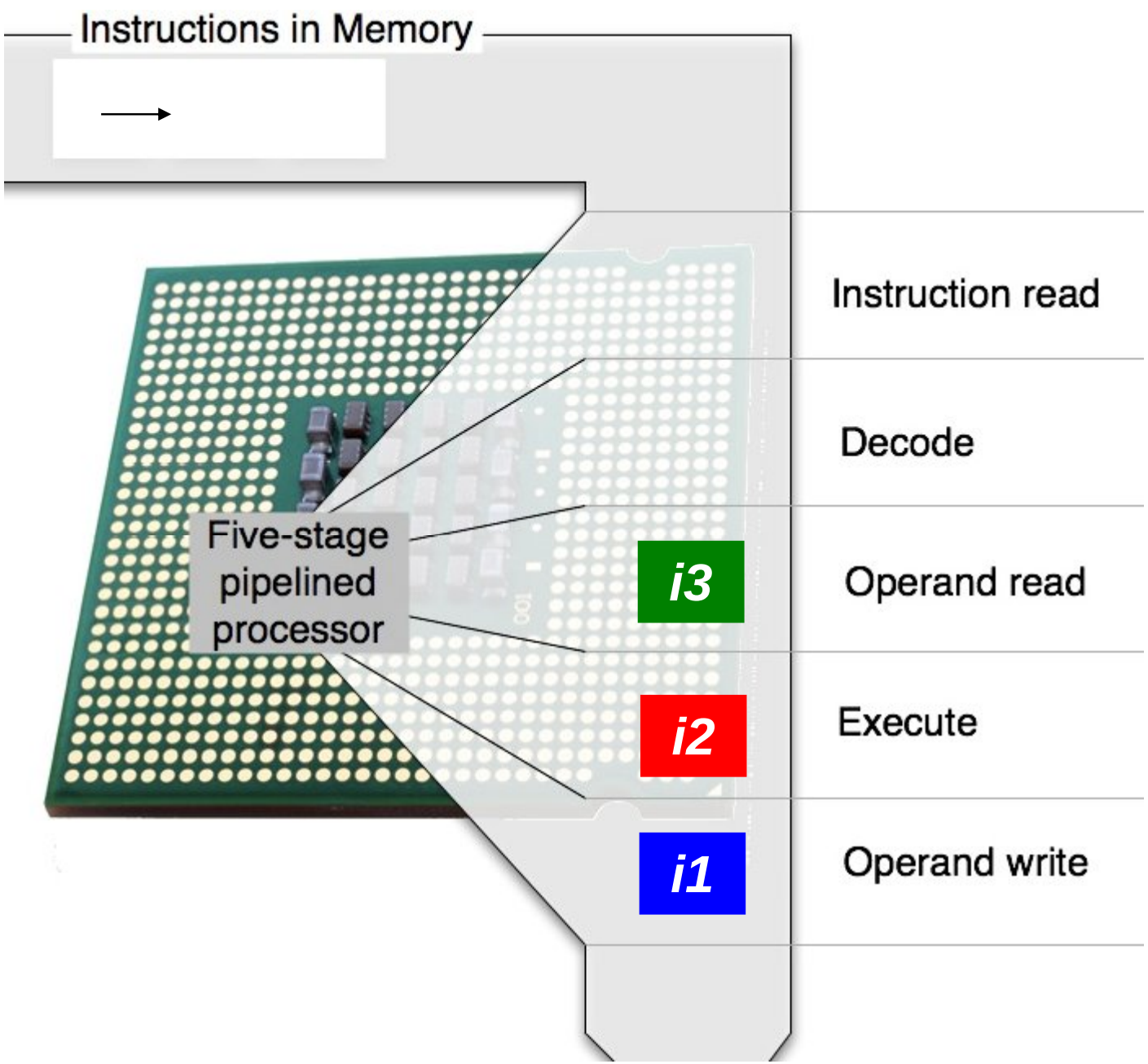

Instructions in Memory
i3
i2
i1
Five-stage
pipelined
processor
Instruction read
Decode
Operand read
Execute
Operand write

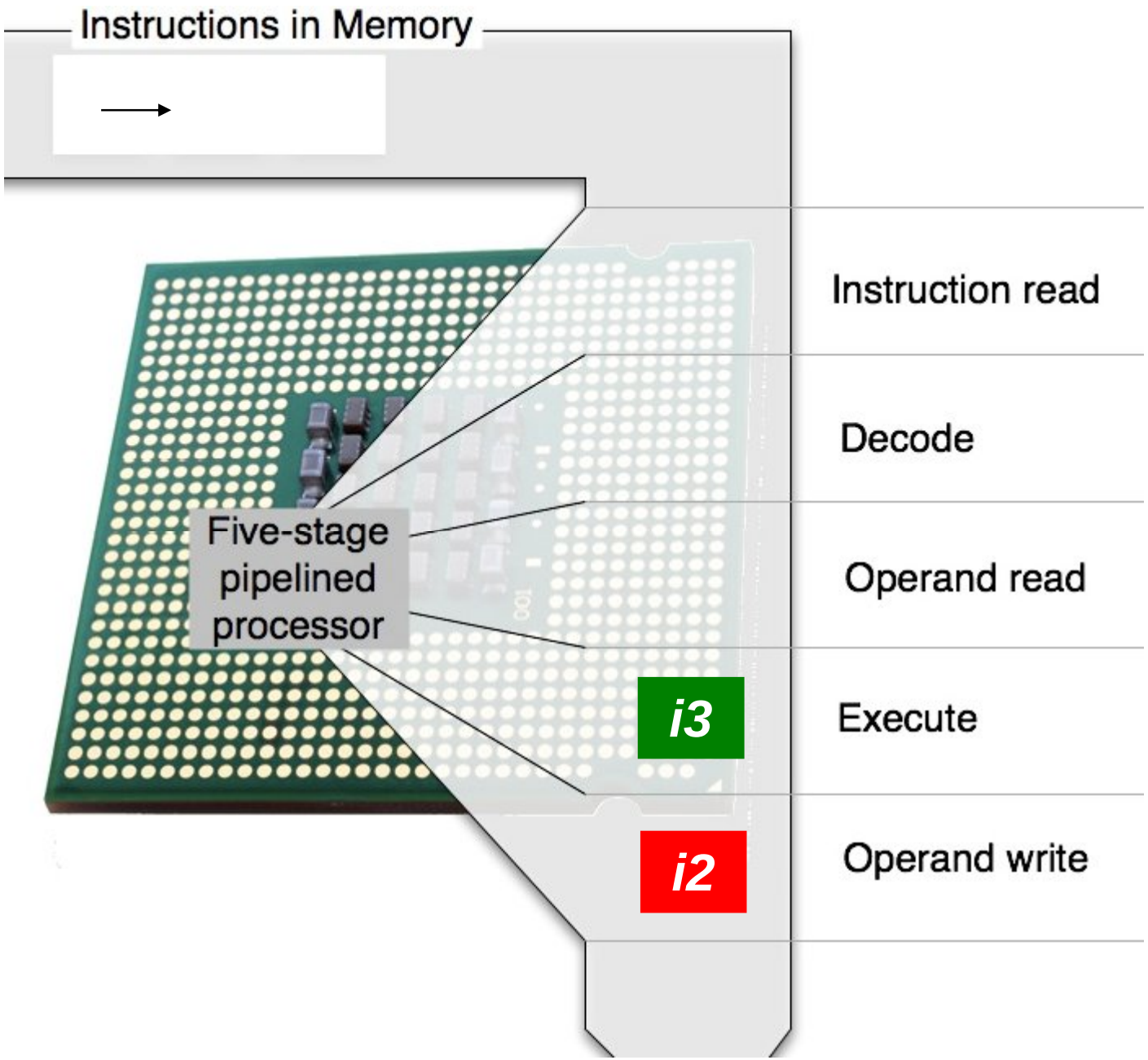

Instructions in Memory
i3
i2
i1
Five-stage
pipelined
processor
Instruction read
Decode
Operand read
Execute
Operand write

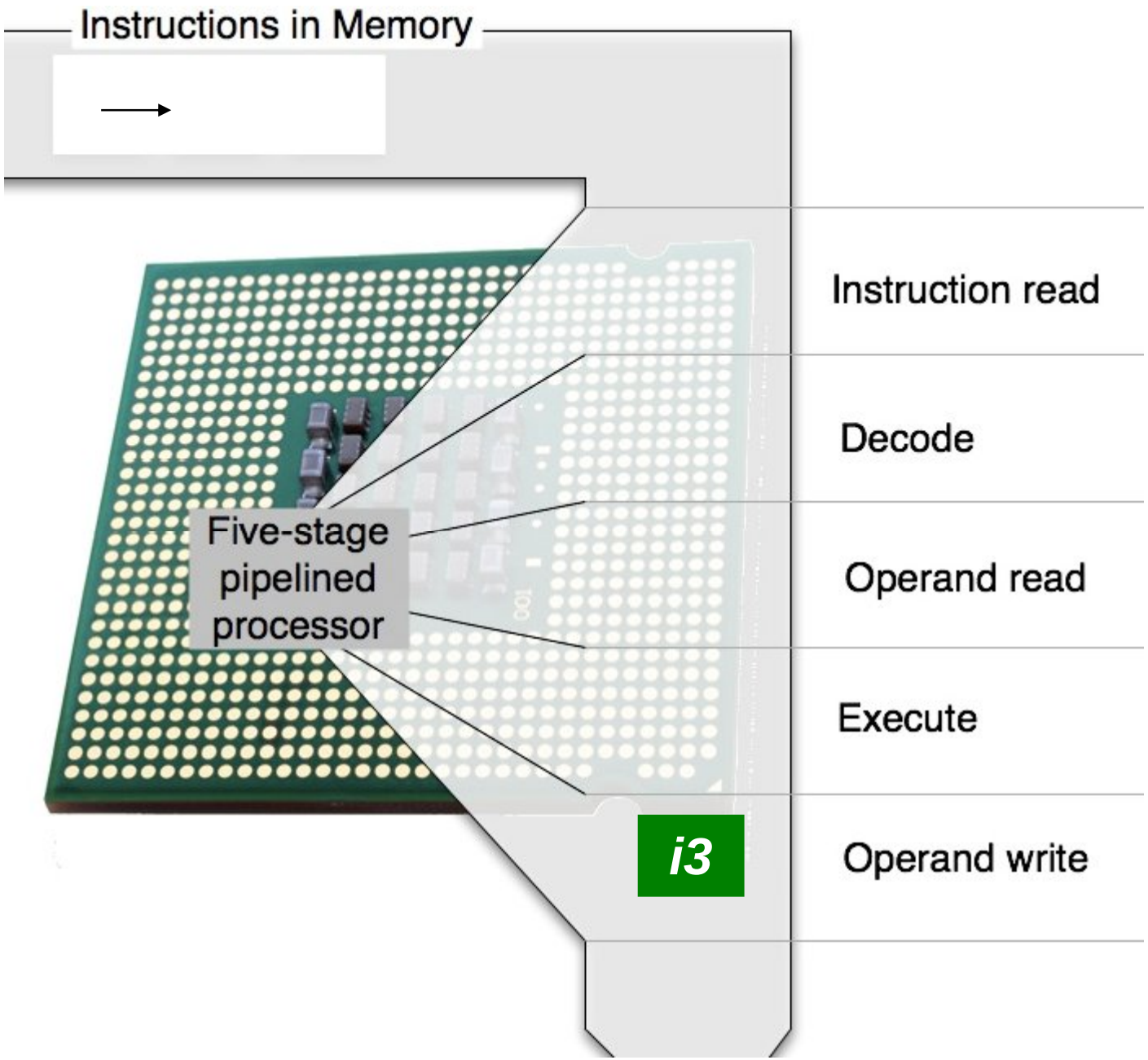

Instructions in Memory
i3
i2
i1
Five-stage
pipelined
processor
Instruction read
Decode
Operand read
Execute
Operand write


Instructions in Memory
i3
i2
i1
Five-stage
pipelined
processor
Instruction read
Decode
Operand read
Execute
Operand write


Instructions in Memory
Five-stage pipelined processor
i3
i2
i1
Instruction read
Decode
Operand read
Execute
Operand write


Instructions in Memory
Five-stage pipelined processor
i3
i2
i1
Instruction read
Decode
Operand read
Execute
Operand write


Instructions in Memory
Five-stage pipelined processor
i3
i2
Instruction read
Decode
Operand read
Execute
Operand write


Instructions in Memory
Five-stage pipelined processor
i3
Instruction read
Decode
Operand read
Execute
Operand write