Chapter 3

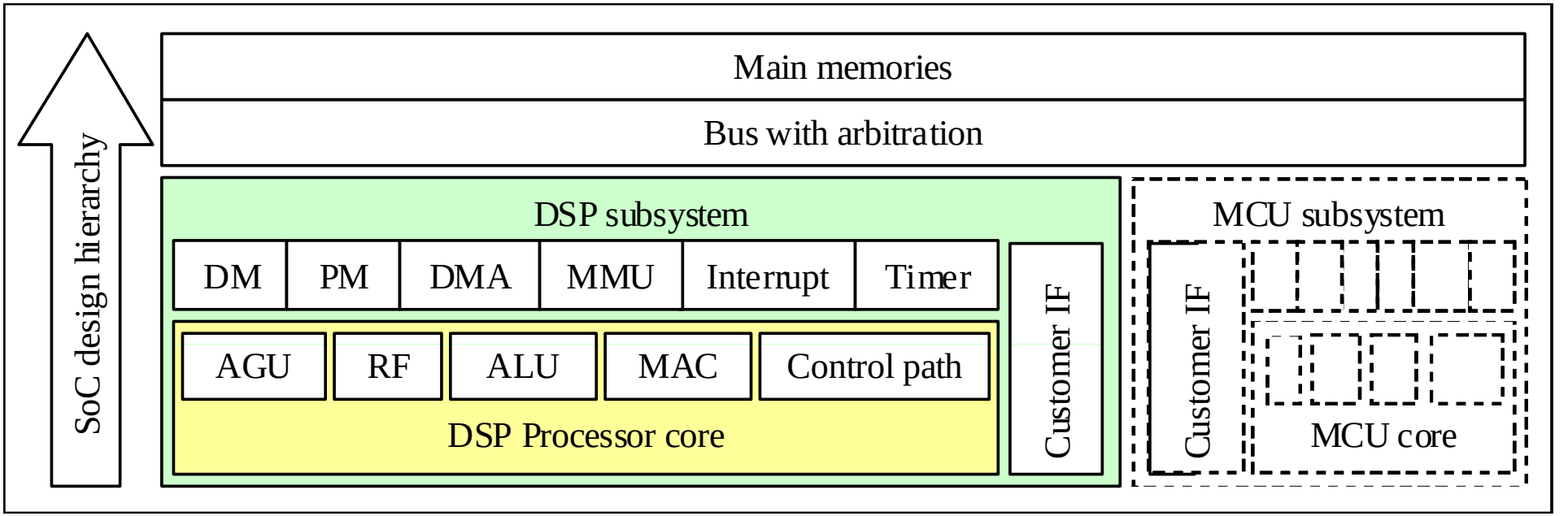
Architectures

Contents

- DSP subsystem and arch in general
- Inside a DSP core
- The difference between DSP and MPU and MCU
- Advanced DSP architectures

DSP subsystem

System on a chip

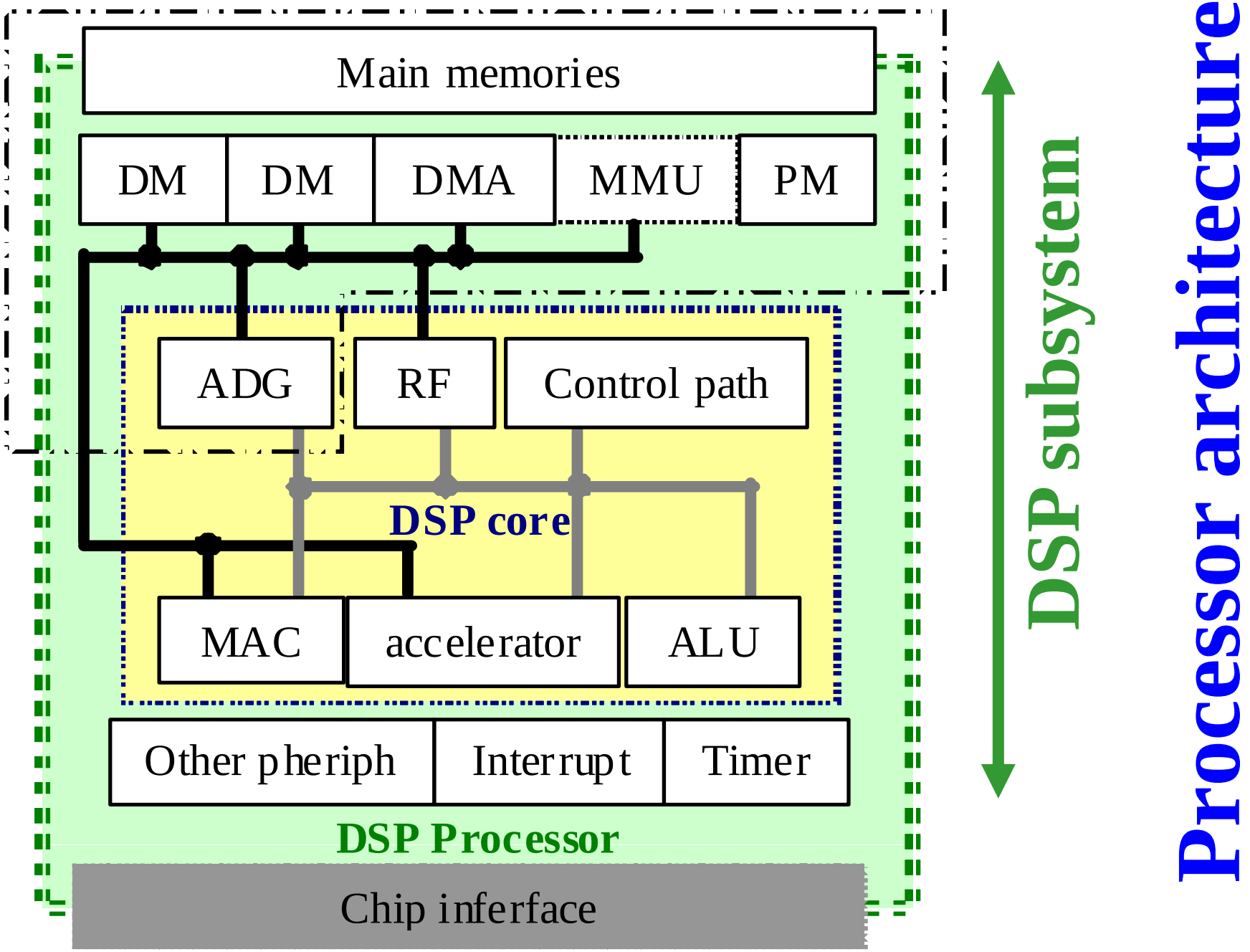


What is a DSP core

What is a DSP processor

What is a DSP subsystem?

Processor architecture



DSP subsystem

- DSP core
- Memory subsystem
- Peripheral subsystem

DSP core

- The core can be divided into three parts:
  - Datapath
  - control path
  - address generation unit (AGU).
- The core components are organized around two data busses:
  - The memory bus is distributed between the core and the memory subsystem.
  - The register bus connects the register file to all units in the core.

Memory subsystem

- The memory subsystem consists of
  - data memories (DM),
  - program (code) memory (PM),
  - AGU, DMA, and MMU.

Peripheral subsystem

- Two levels
  - DSP core peripherals
  - DSP peripherals
- Two types
  - Functional peripherals : access or configured by programs
  - Non-functional peripherals: JTAG port

DSP core peripherals

- Timers for counting clock cycles and events
- Interrupt controller for handling interrupts
- DMA (Direct Memory Access) controller for handling data transfers to/from main memory and between other memories/ports
- MMU (Memory Management Unit) for reliable and efficient (address space) memory usage

DSP peripherals

- Host interface for connecting an MCU
- Data ports for connecting off-chip memory
- Serial interfaces

Architecture and microarchitecture

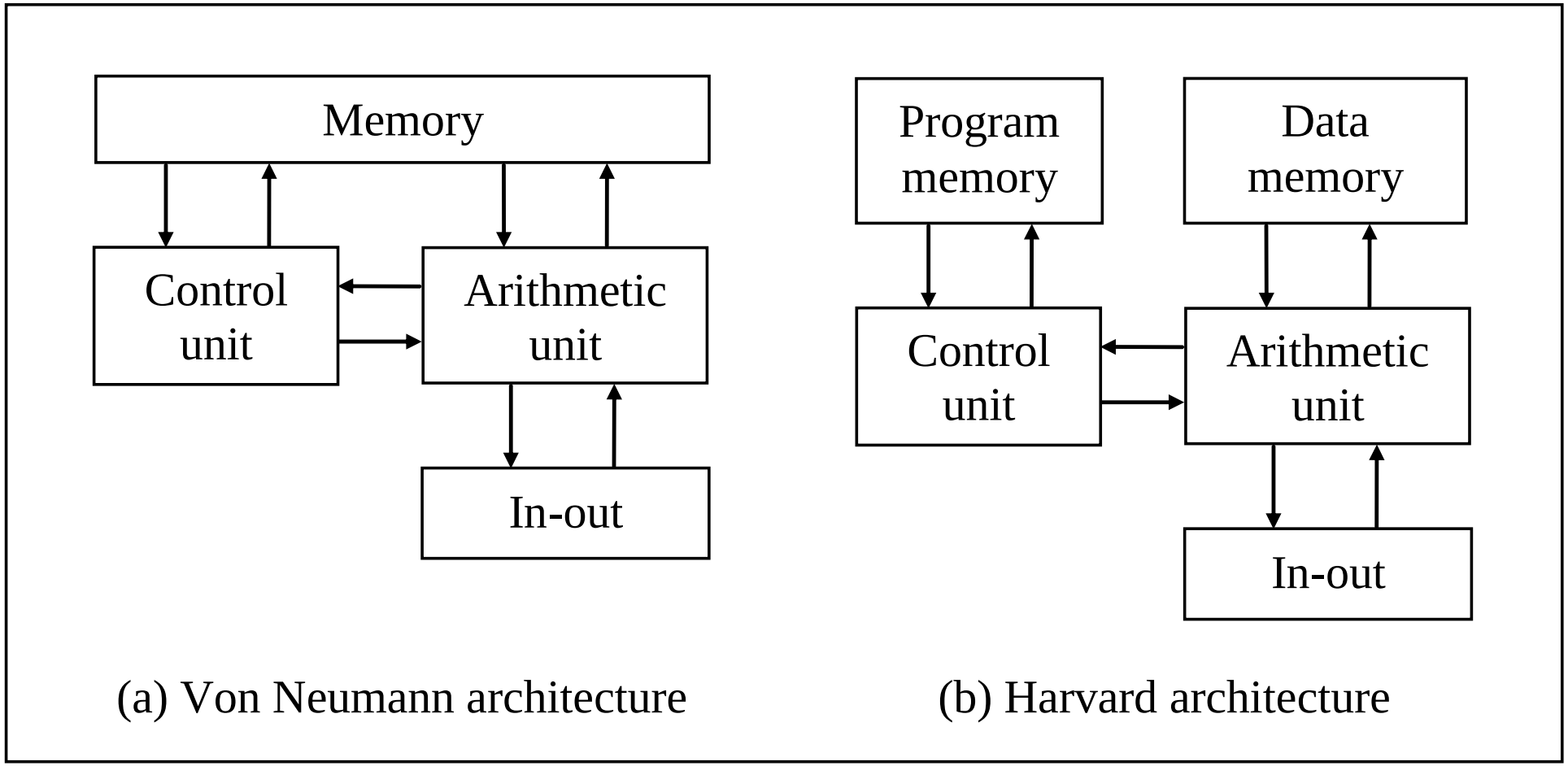
- The processor architecture is the hardware organization of the core and its peripherals including the memory bus architecture. Architecture represents relations of modules
- The microarchitecture design is the specification of functional modules
- ASIP microarchitecture design is the implementation of an ISA specification into hardware modules.

Microarchitecture design

1. Expose and allocate all microoperations from each assembly instruction to HW modules
2. Specify the pipeline and allocates each microoperation to one or several pipeline stages
3. Specify the HW multiplexing
4. Specify I/O and intermodule operations
5. Optimize the proceeding four steps

DSP architecture

History of DSP architectures

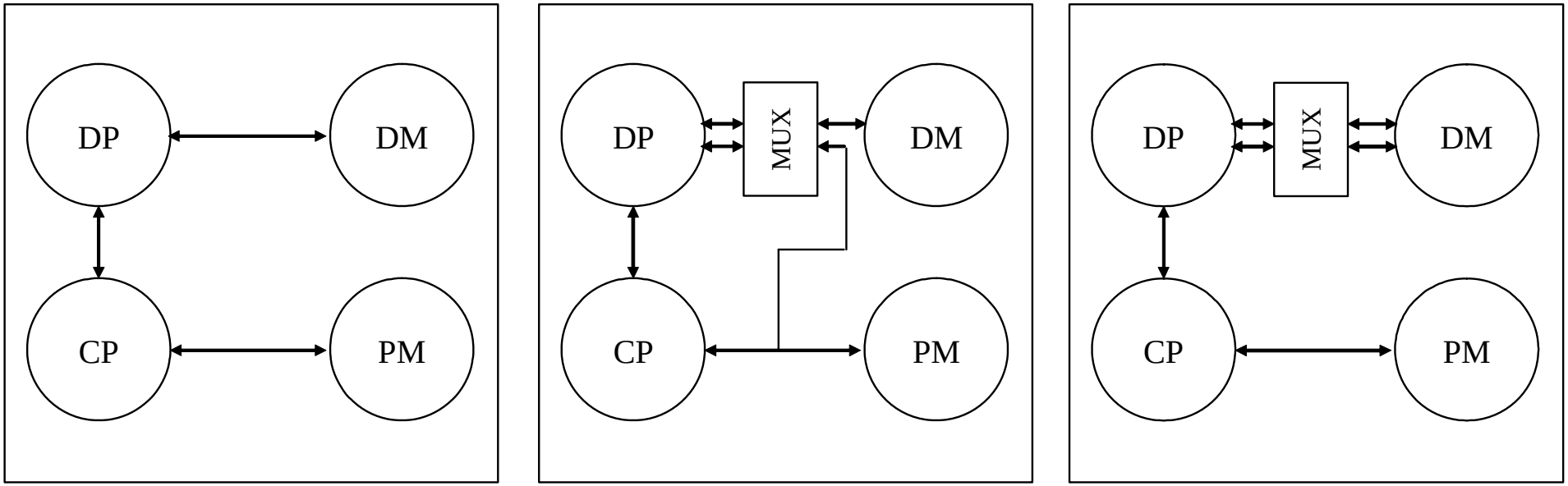


(a) Von Neumann architecture

(b) Harvard architecture

History of DSP architectures

A convolution step requires a data and a coeff



(a)

A convolution tap requires multiple cycles

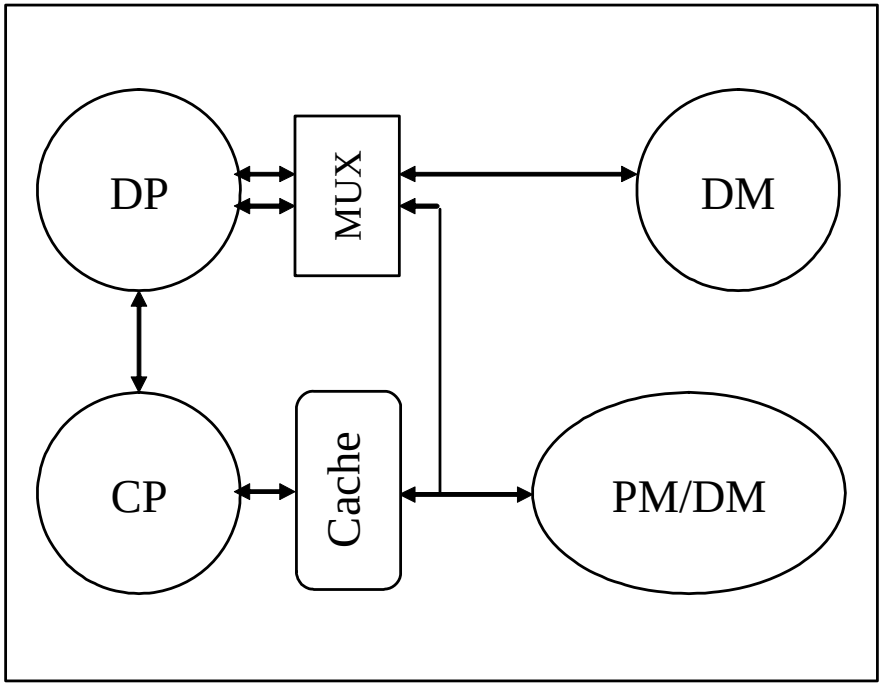
(b)

Supporting a HW loop convolution;
No iteration support

(c)

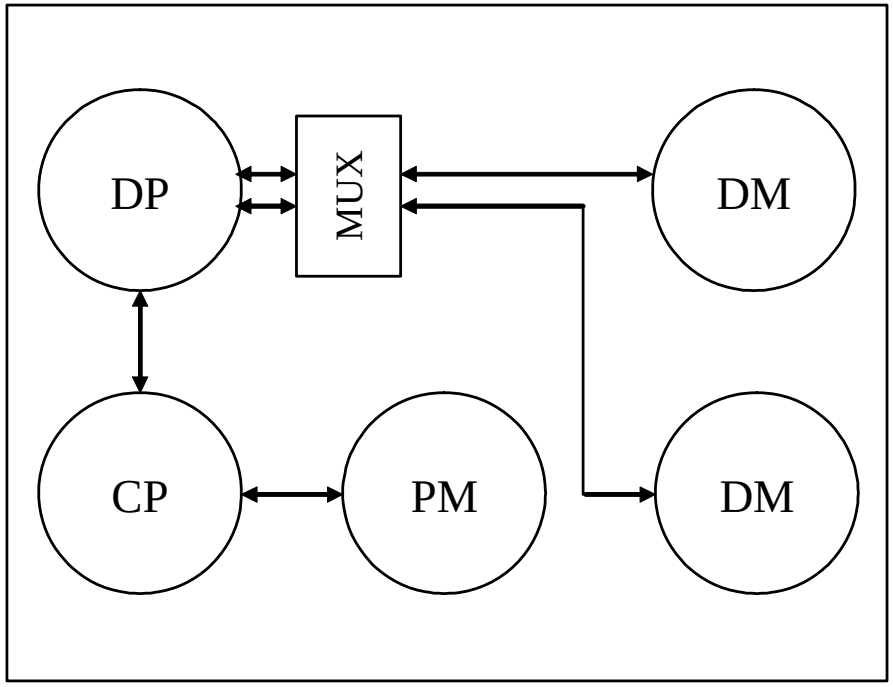
Dual port memory
Slow memory accesses

History of DSP architectures



(d)

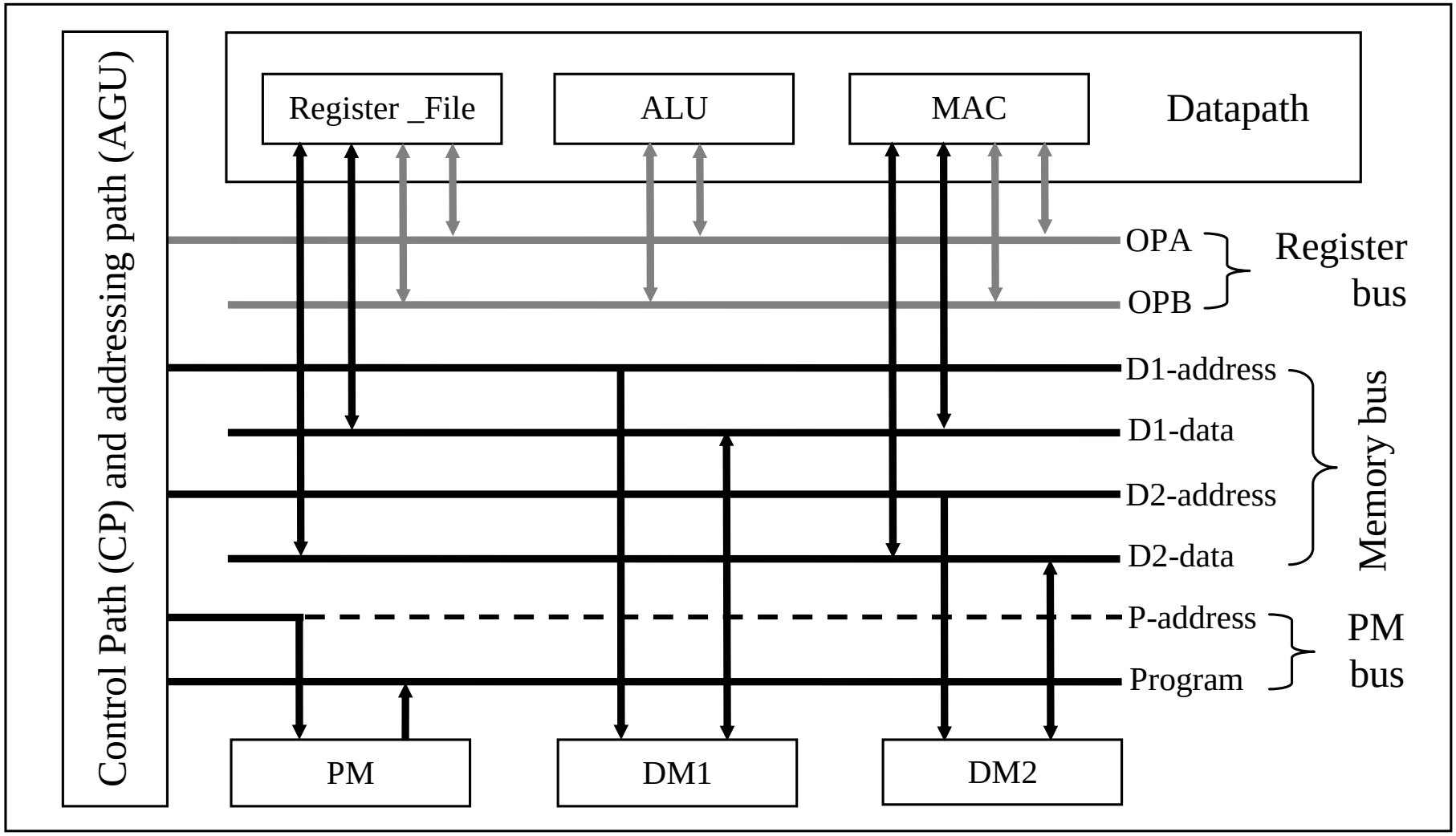
Small cache;
Supporting a HW loop
with multiple instructions



(e)

More flexible;
Not limited by the size of cache

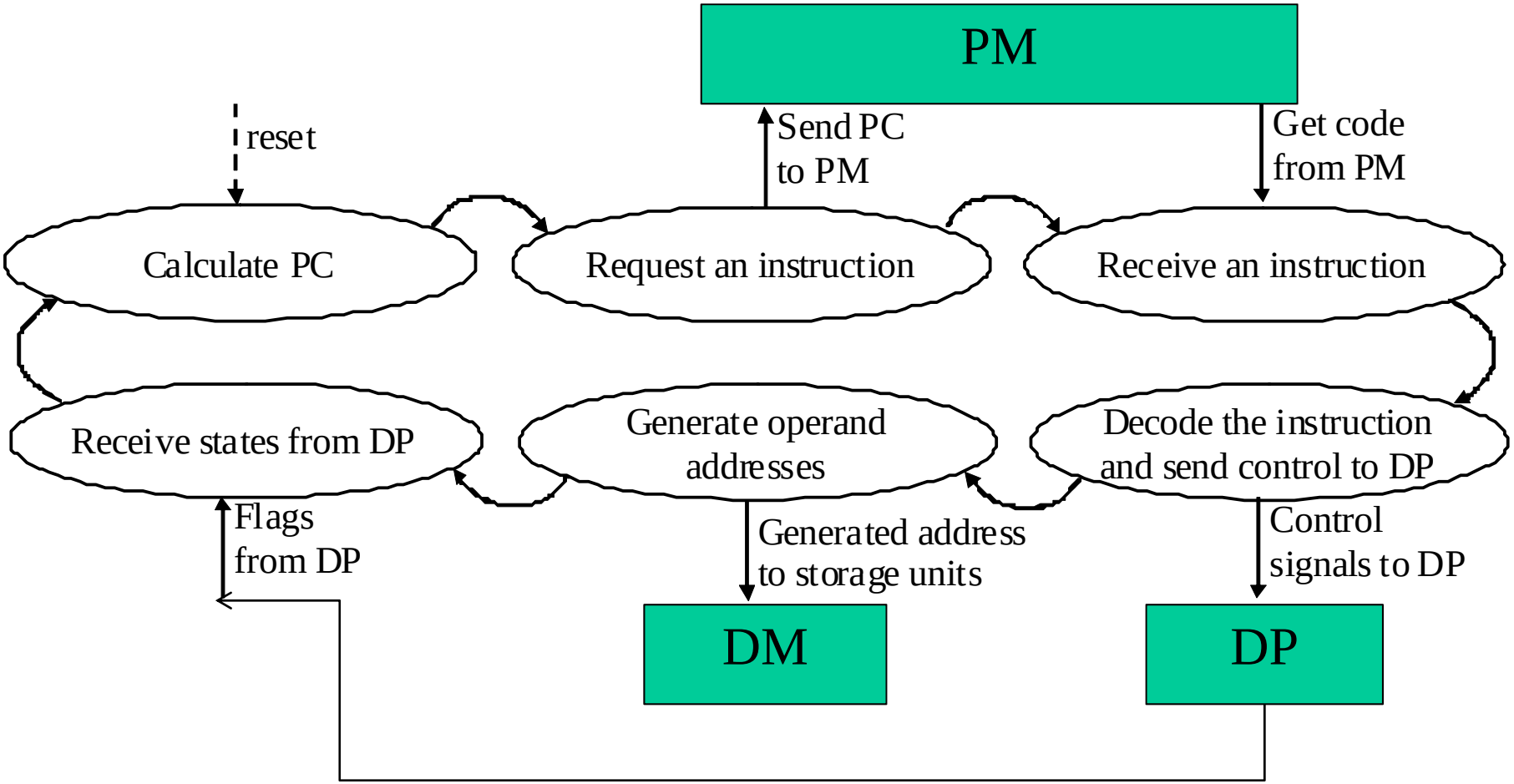
A typical DSP bus architecture



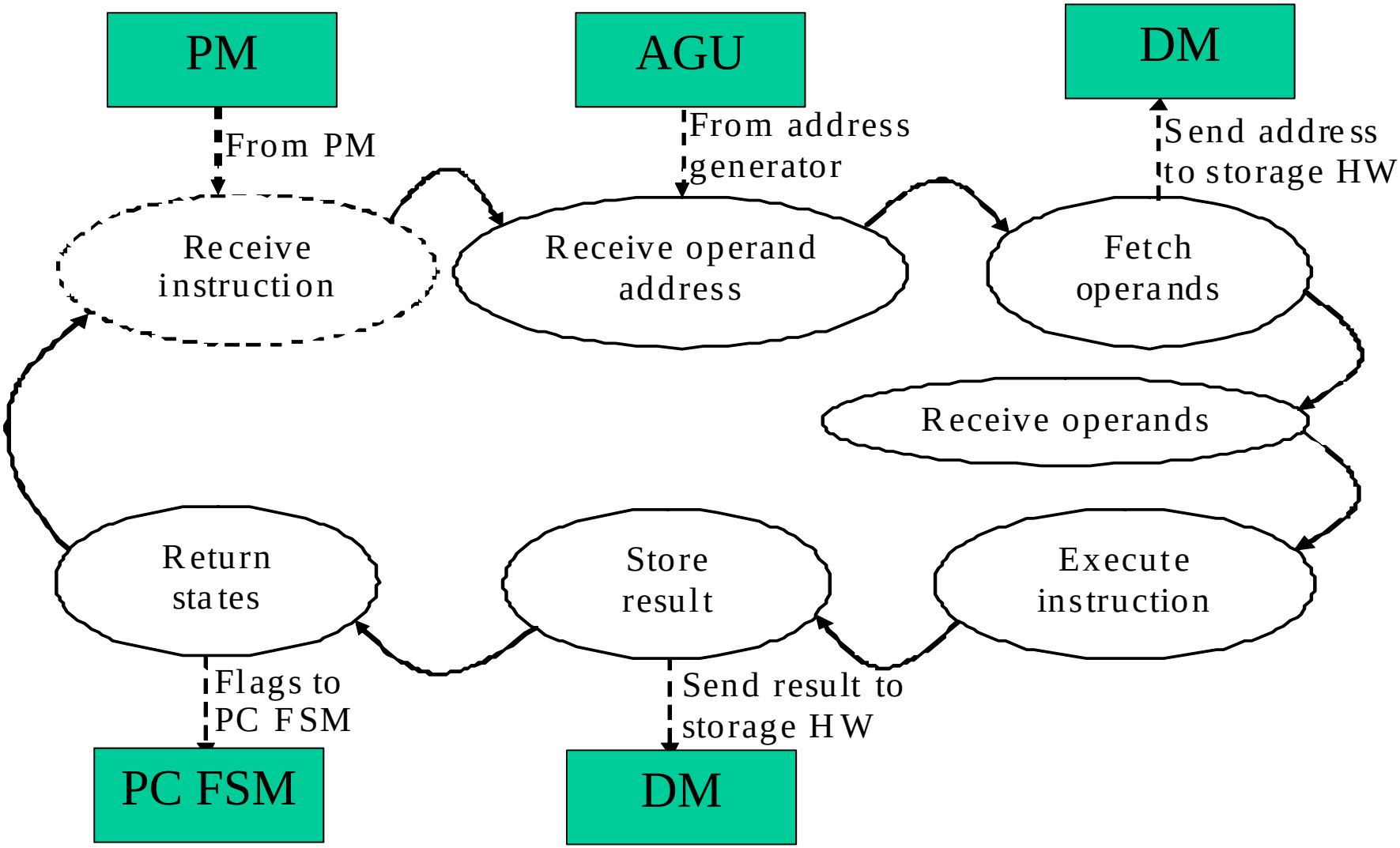
In DSP

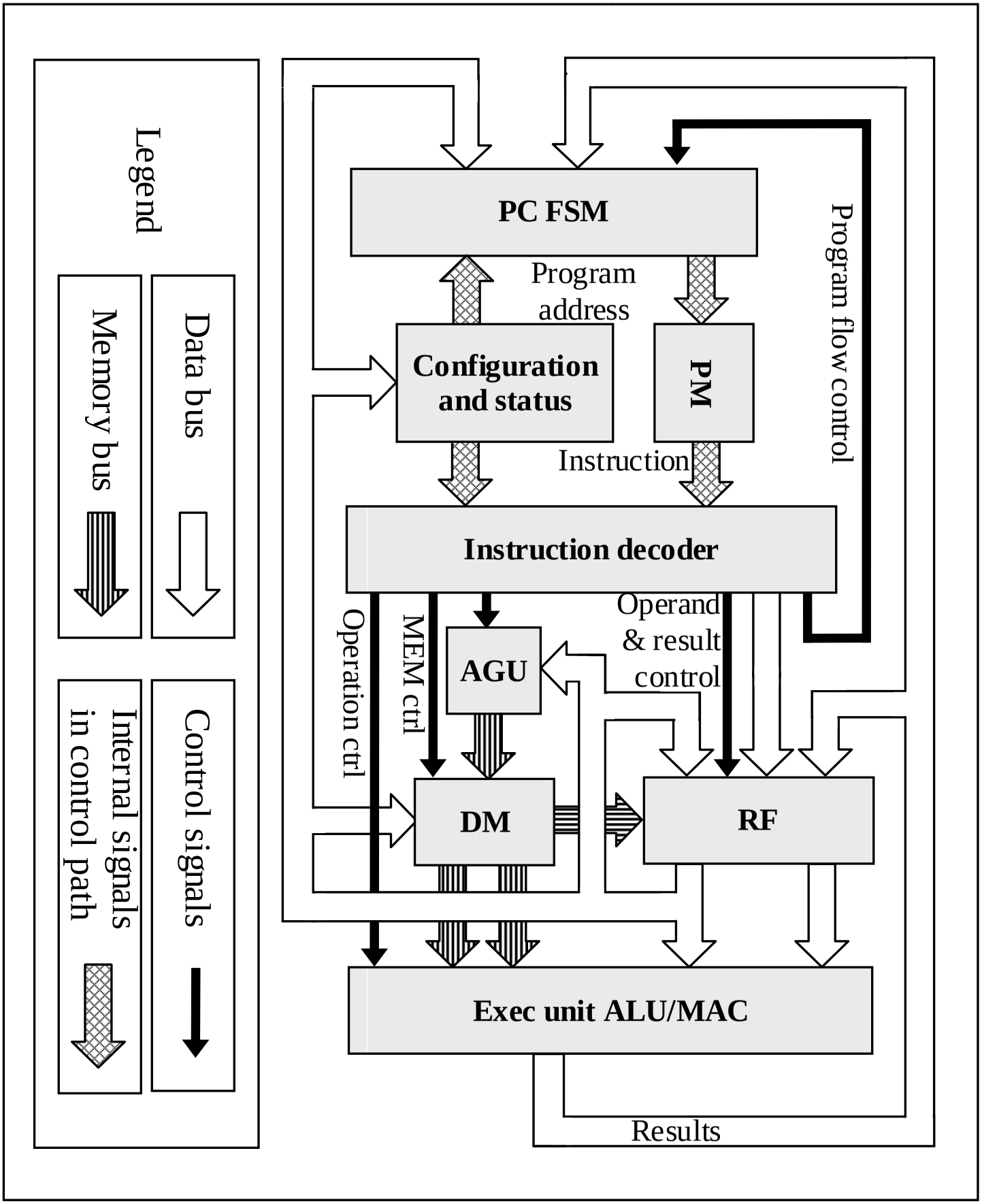
- Two state machines
  - Instruction flow FSM: calculate the address of the instruction to be fetched
    - Decode the fetched instruction
    - Calculate the address of each operand
    - Get flags from DP
  - Data flow FSM: execution the decoded instruction
    - Fetch operands
    - Store results
    - Send flags to instruction flow FSM

Control flow FSM



Data flow FSM





A complete view of a DSP processor

Processor architecture design

- 1. Specifies modules and interconnects between modules as well as input/outputs according to the requirements from the instruction set.
- 2. Specifies module parameters such as memory address space, register file size, data types, etc.

How to specify top level bus

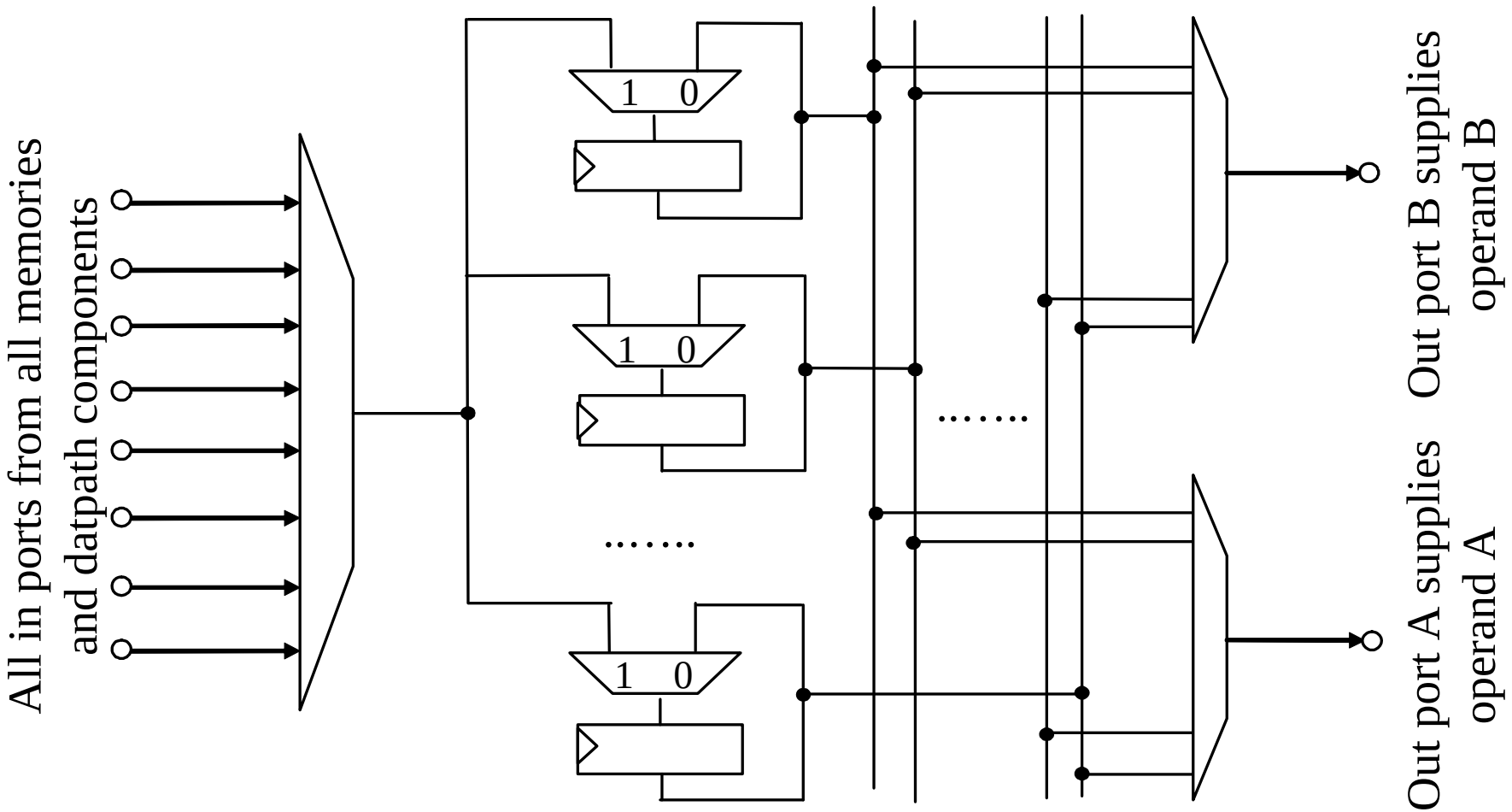
- All operands and results following all instructions

Table 3.6

From memory	M1→MAC	M2→MAC	M1→RF	M2→RF		
To memory	RF→M1	RF→M2	AGU→M1	AGU→M2		
From register	RF→MAC	RF→MAC	RF→ALU	RF→ALU	RF→CP	RF→AGU
To register	M1→RF	M2→RF	MAC→RF	ALU→RF	CP→RF	
Others	CP→AGU	CP→ALU	CP→RF			

Specify each module by arranging its inputs and outputs (Table 3.5)
Similarly, specify the connections among the modules (Table 3.6)

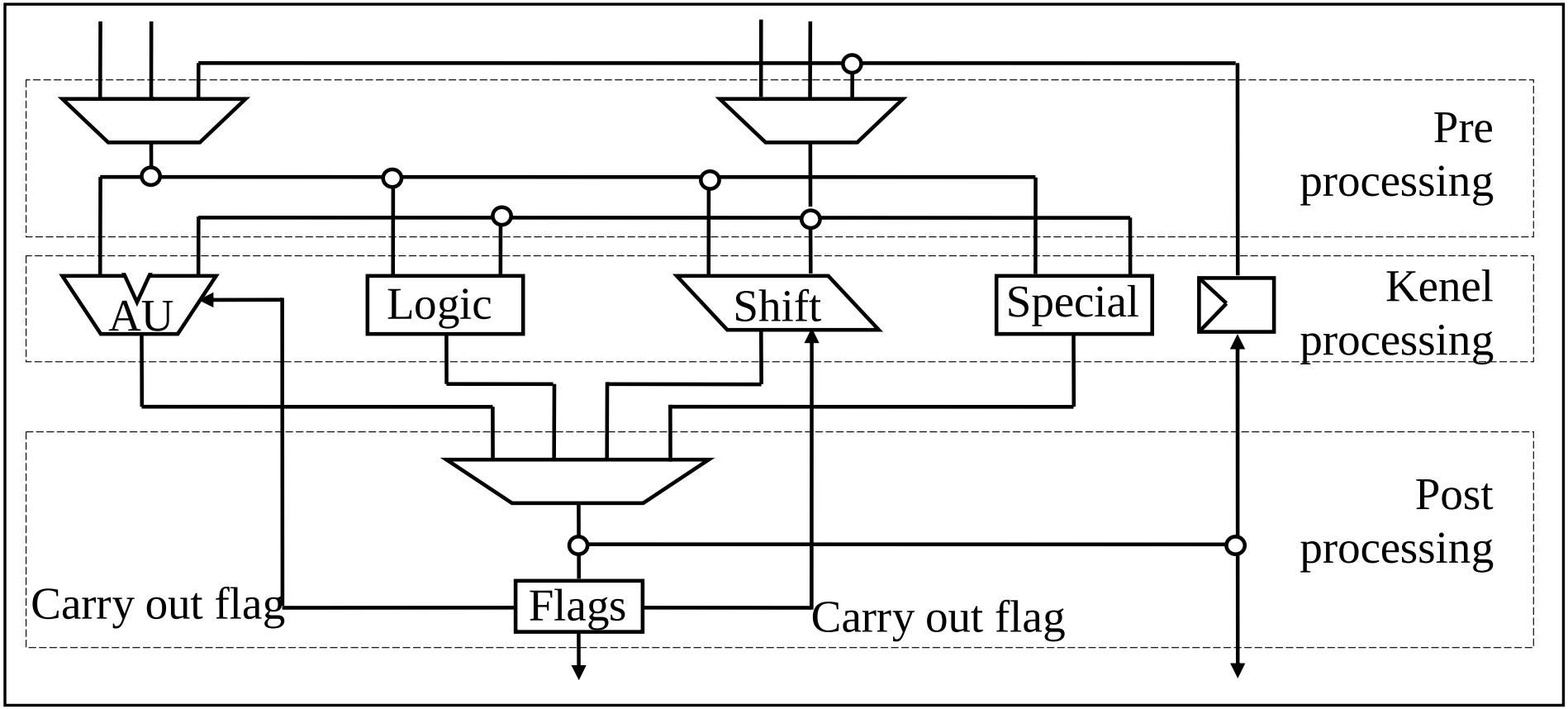
Register file



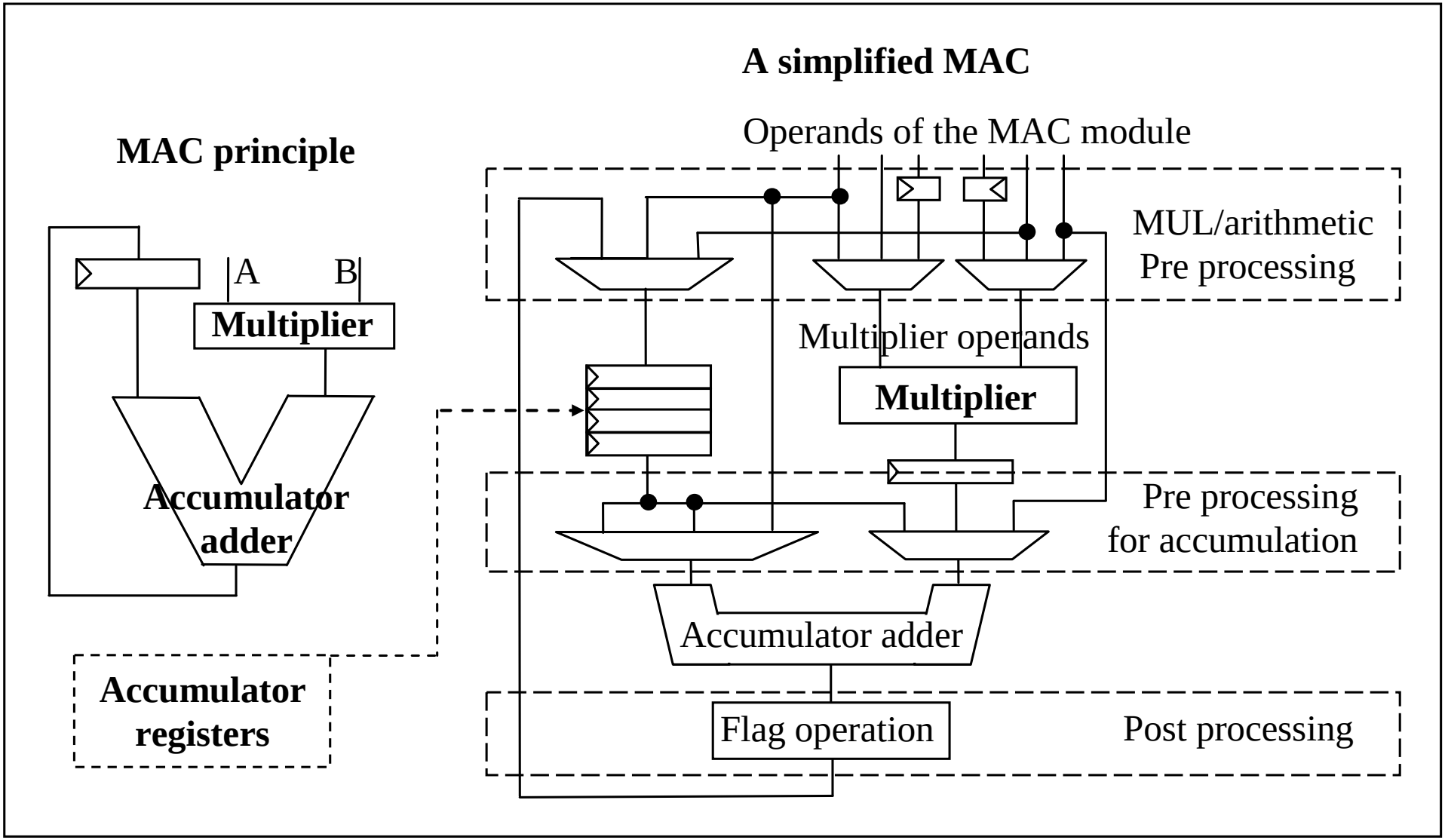
ALU

Preprocessing: operands selection & preprocessing

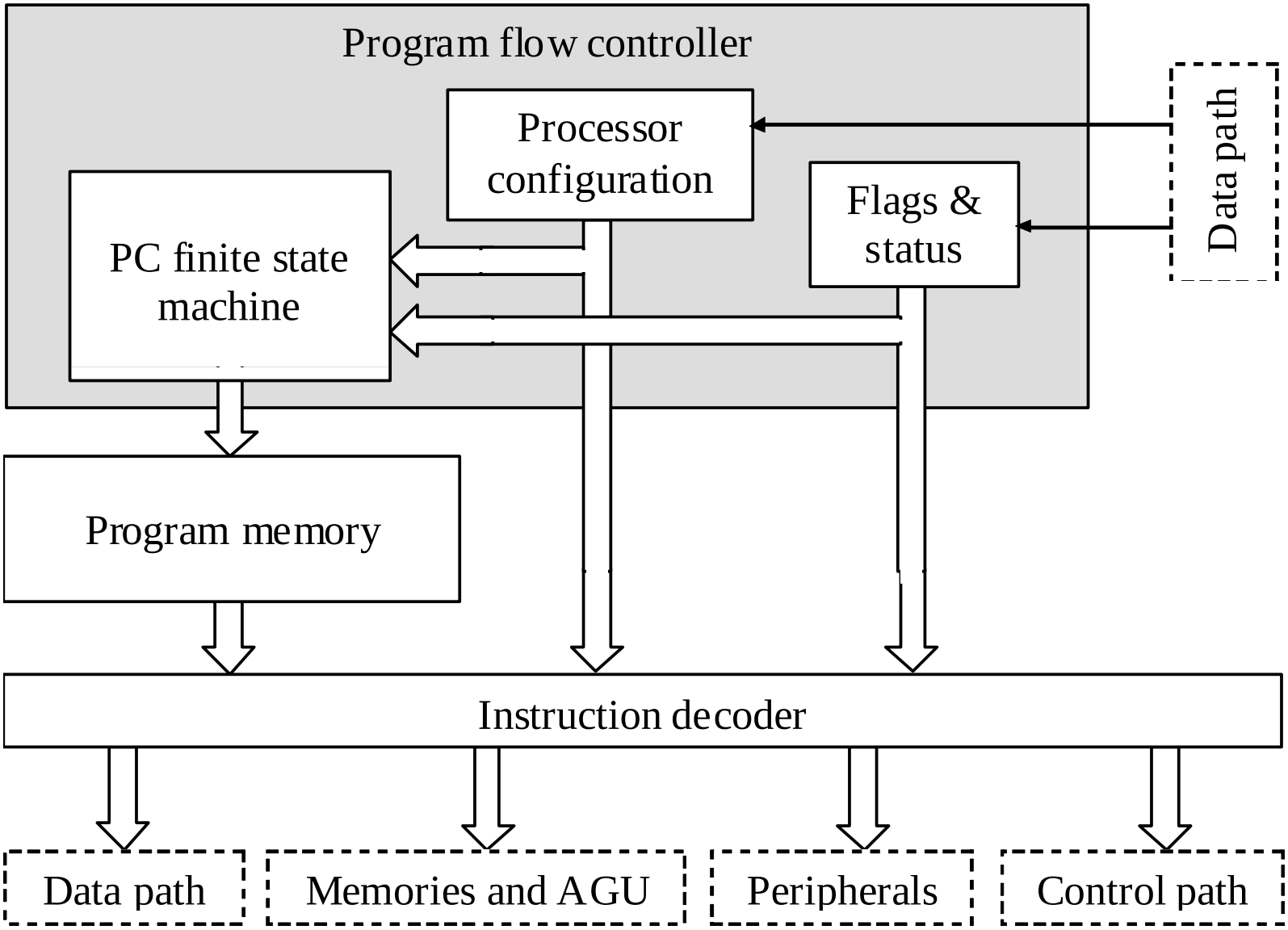
Postprocessing: result selection & flag computing



MAC Unit


MAC principle
A
B
Multiplier
Accumulator adder
Accumulator registers
A simplified MAC
Operands of the MAC module
MUL/arithmetic Pre processing
Multiplier operands
Multiplier
Pre processing for accumulation
Accumulator adder
Flag operation
Post processing

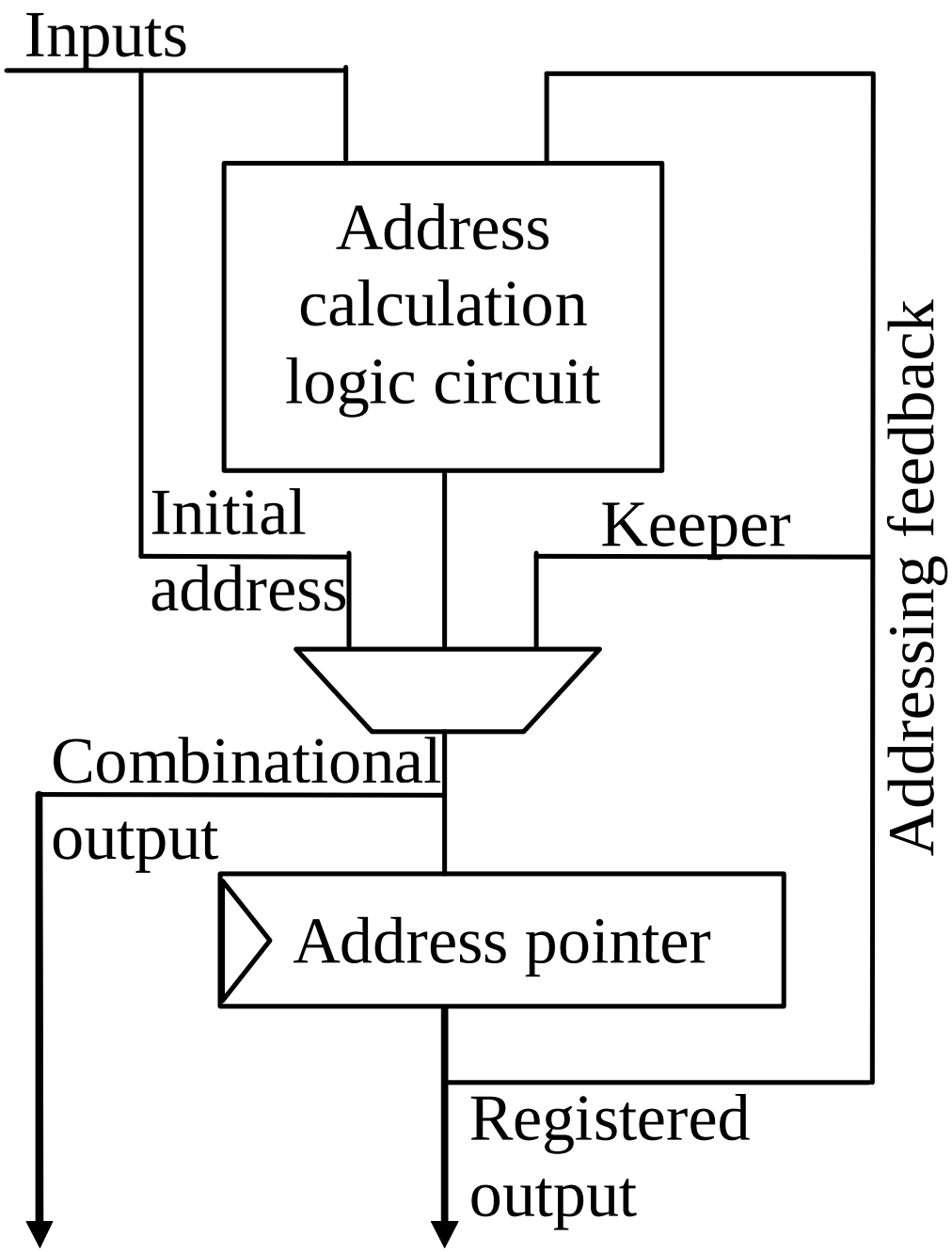
Control path



Pipelining

- Partition instruction execution into multiple steps
  - Enhance the throughput
  - Equi-partition
- Allocate each of the step to independent hardware
- Assign each of them to a pipeline stage
- Run all operations in the pipeline stage in parallel for different instructions
  - Branch penalty

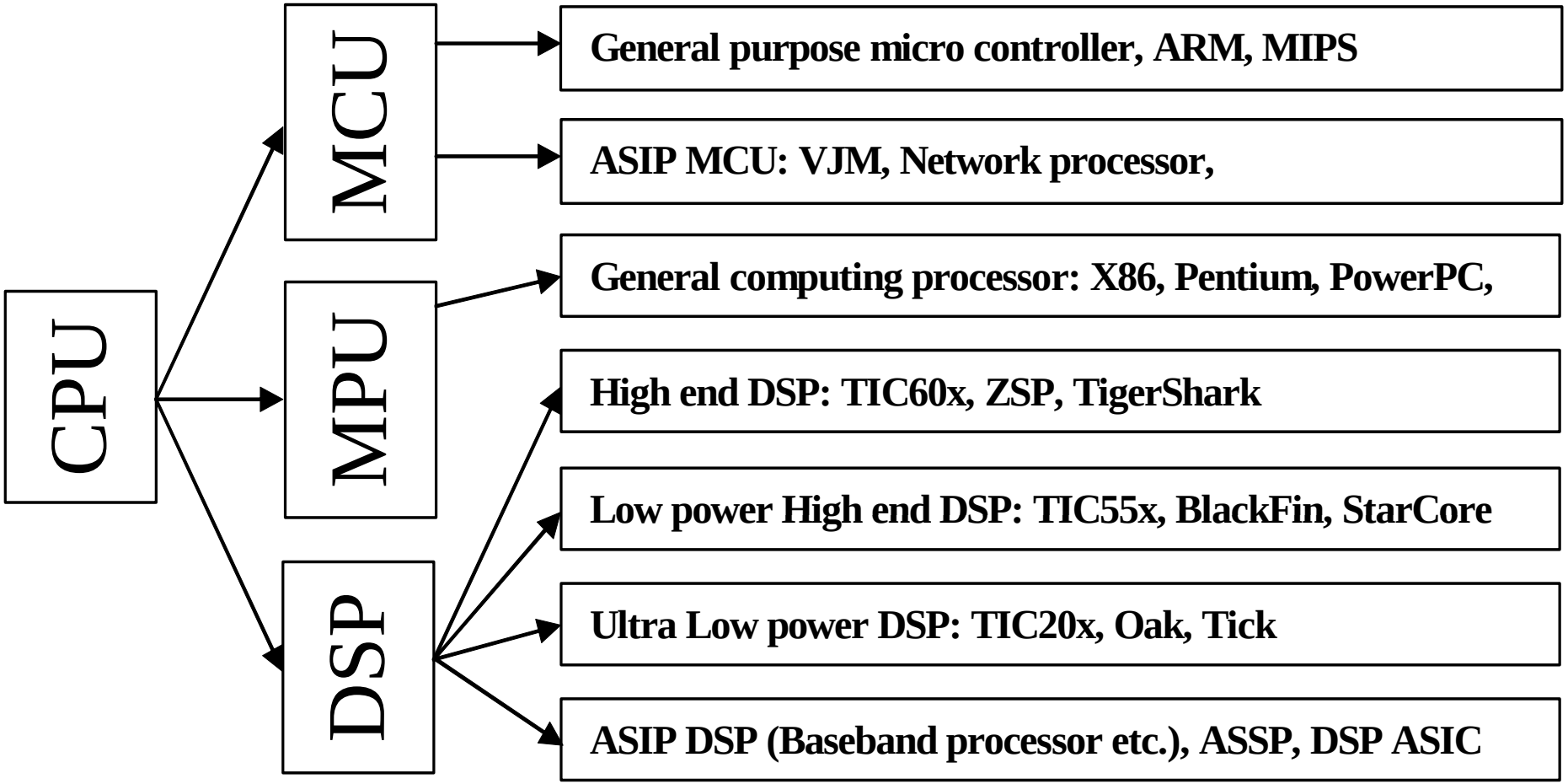
Address Generator



Differences between design of DSP and MPU

- The MPU (GPP) designers think of performance and flexibility as well as the compiler-friendly instruction set.
  - Application: general, unknown
- The ASIP DSP designers think of application and cost first, and the challenge is to be efficient.
- The goal of an ASIP design is to obtain higher performance for a given silicon area, or a given power consumption, or a given cost limitation.

Differences between DSP, MPU, MCU



Advanced architectures

- Possibilities of parallel DSP
  - Specific applications
  - Simple scheduling
  - Vector based algorithms and processing
- Challenges of DSP parallelization
  - Hard real time
  - High performance
  - Data and control dependencies

Two kinds of parallelization

- Pipelined parallelization for streaming signal processing. Several processing tasks/units are connected into a chain. The output of the first task is naturally the input to the second task.
- Data parallelization for vector signal processing. The parallelism existing in the data is recognized and utilized for parallel computing.

Performance measurement

- IPC, instructions (issued/executed) per clock cycle
- CPI is the inverse of IPC
- Efficiency of an ILP machine is IPC/W
  - W is the width of the ILP,
- OPC, the number of arithmetic instructions per clock cycle
  - MPOS, GOPS
  - That is what you really want

Conclusion

- DSP Architecture is diverging
- General DSP processor is a RISC with CISC features
- Parallel bus to minimize access latency
- Much parallel computing and access features

Conclusion

- DSP subsystem and arch in general
  - Definition, scope, components
- Inside a DSP core
  - RF, ALU, MAC, AGU, and CP
- Advanced DSP architectures