Program Modeling Concepts:

Lesson-7: MODELING OF MULTIPROCESSOR SYSTEMS

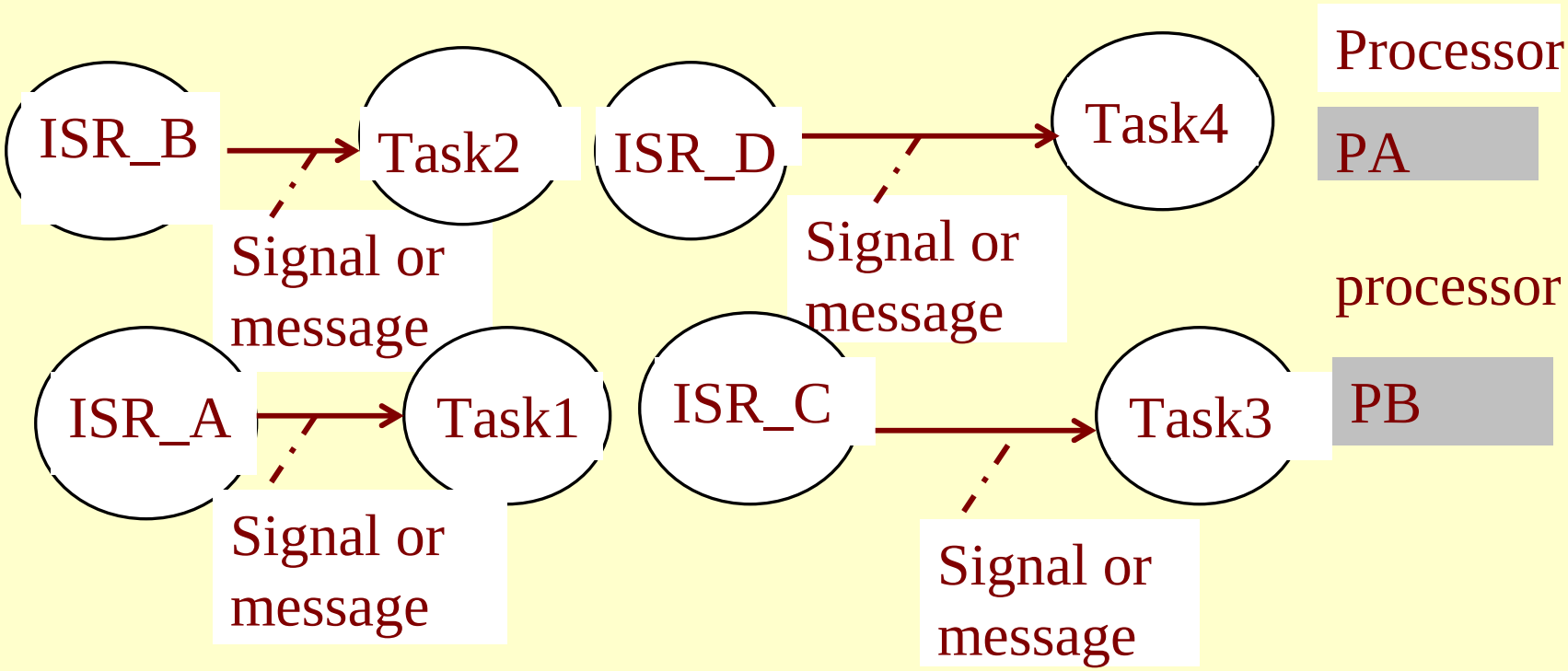
Multiprocessor system

- A multiprocessor system uses two or more processors for faster execution of the (i) Program functions, (ii) tasks or (iii) single instruction multiple data instructions

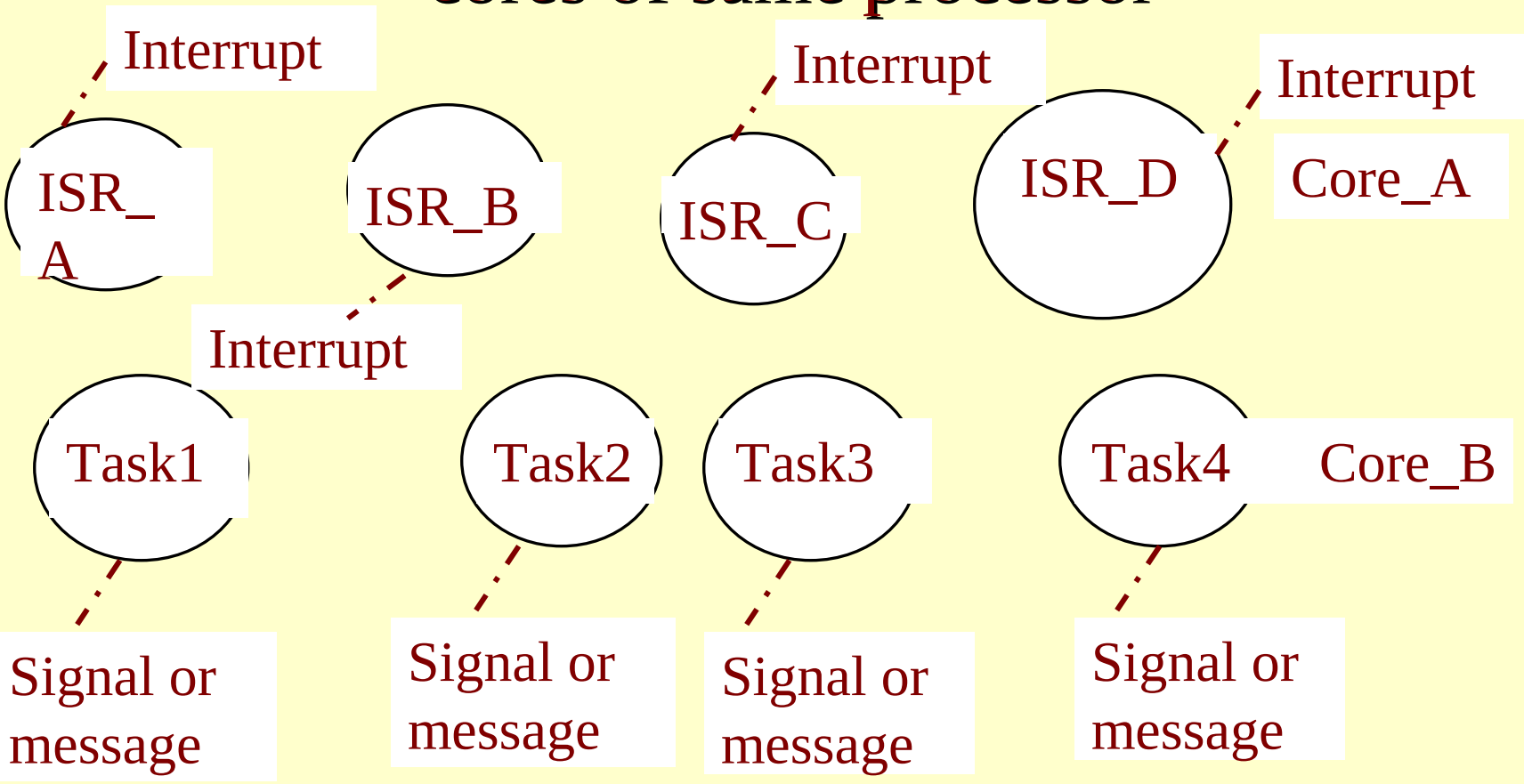
A large complex program

- Partitioned into the tasks or sets of instructions (or processes or threads) and the ISRs.
- The tasks and ISRs can run concurrently on different processors and by some mechanism the tasks can communicate with each other
- Multiple-instructions multiple- data instructions
- Very long instruction words (VLIWs)

Static scheduling of tasks and ISRs on two processors



Static scheduling of tasks and ISRs on two cores of same processor



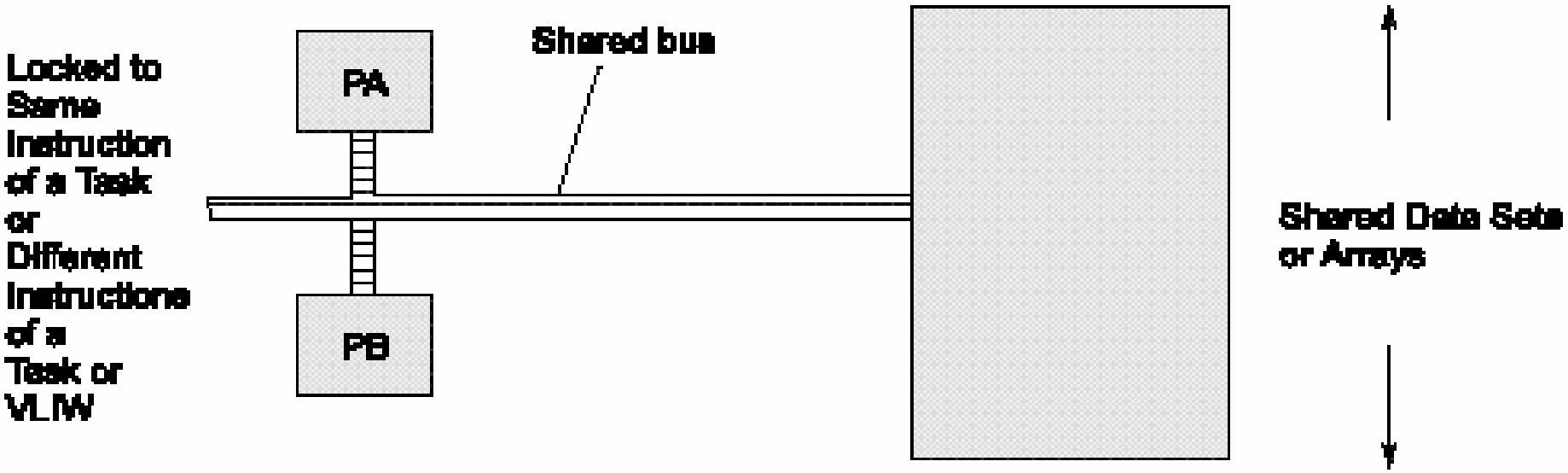
Static scheduling

- Means that a compiler compiles such that the codes are run on different processors or processing units as per the schedule decided and this schedule remains static during the program run even if a processor waits for the others to finish the scheduled processing

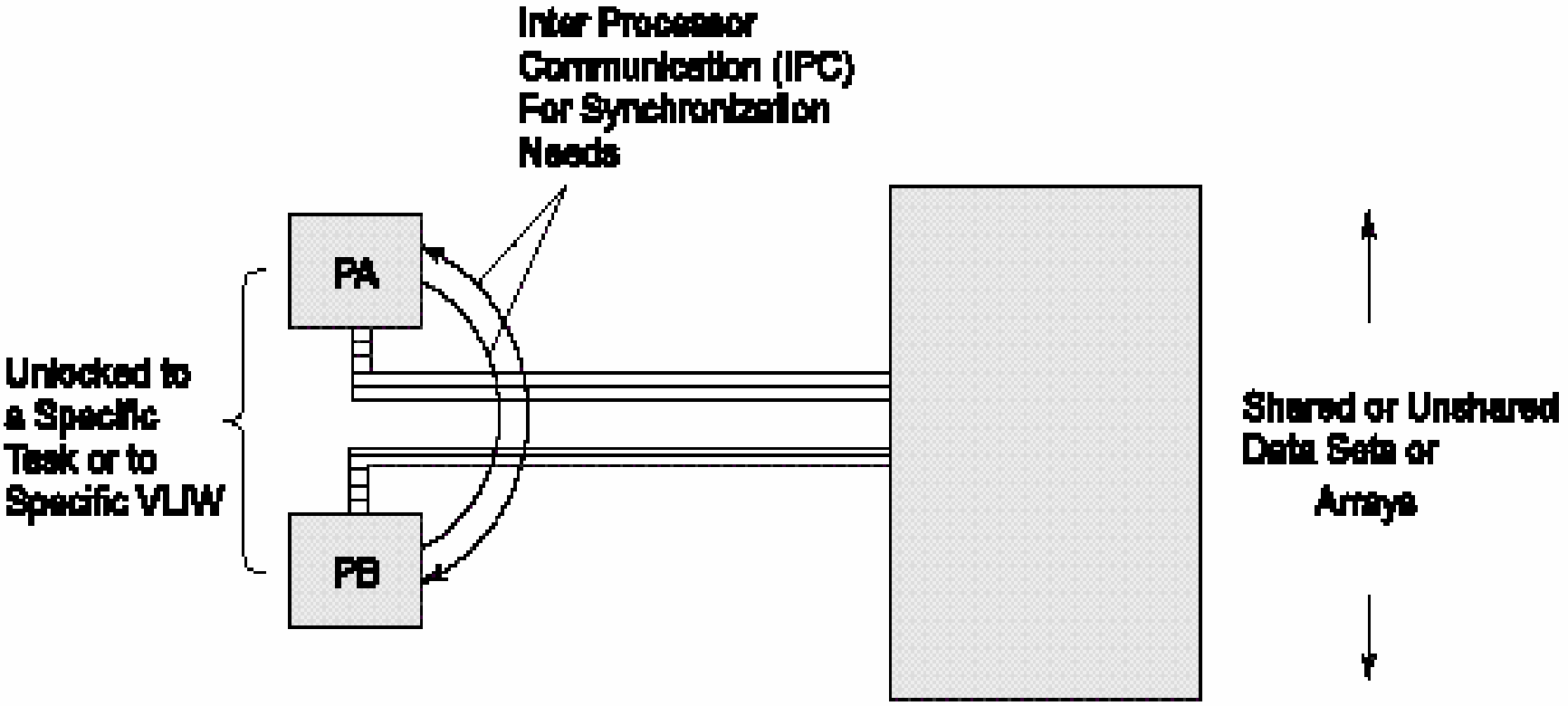
The Multiprocessor System memory-interface Models

- Share the same address space through a common bus (tight coupling)
- Processors have different autonomous address spaces (like in a network) as well as shared data sets and arrays, called loose coupling .

Tight Coupling of two processor to a common memory



Loose Coupling of two processor to a common memory



The Multiprocessor System Programming Model Needs

- Partition the program into tasks or sets of instructions between the various processors
- Schedule the instructions over the available processor times and resources so that there is optimum performance.

The Multiprocessor System Programming Model Needs

- Partition of processes, instruction sets and instruction(s)
- Schedule the instructions, SIMDs, MIMDs, and VLIWs within each process and scheduling them for each processor
- Concurrent processing of processes on each processor

The Multiprocessor System Programming Model Needs

- Concurrent processing on each superscalar unit and pipeline in the processor
- Static scheduling by compiler, analogous to the scheduling in a superscalar processor.

The Multiprocessor System Programming Model Needs

- Hardware scheduling, for example, whether static scheduling of hardware (processors and memories) is feasible or not [It is simpler and its use depends on the types of instructions when it does not affect the system performance.]

The Multiprocessor System Programming Model Needs

- Static scheduling issue [For example, when the performance is not affected and when the processing actions are predictable and synchronous.]

The Multiprocessor System Programming Model Needs

- Synchronising issues, synchronisation means use of inter -processor or process communications (IPCs) such that there is a definite order (precedence) in which the computations are fired on any processor in multiprocessor system.

The Multiprocessor System Programming Model Needs

- Dynamic scheduling issues [For example, when the performance is affected when there are interrupts and when the services to the tasks are asynchronous. It is also relevant when there is preemptive scheduling as that is also asynchronous.]

Methods of scheduling and synchronising the execution of instructions, SIMDs, MIMDs, and VLIWs

- Scheduling is done after analyzing the scheduling and synchronising options for the concurrent processing and scheduling of instructions, SIMDs, MIMDs and VLIWs

Method 1 of concurrent processing

- Schedule each task so that it is executed on different processors and synchronise the tasks by some inter -processor communication mechanism

Method 2 of concurrent processing

- when an SMID or MIMD or VLIW instruction has different data (for example, different coefficients in Filter example
- Each task is processed on different processors (tightly coupled processing) for different data.
- This is analogous to the execution of a VLIW in TMS320C6, a recent Texas Instruments DSP series processor.
- TMS320C6 employs two identical sets of four units and a VLIW instruction word can be within four and thirty-two bytes.

Method of concurrent processing

- TMS320C6 has instruction level parallelism when a compiler schedules such that the processors run the different instruction elements into the different units in parallel.
- The compiler does ***static scheduling for VLIWs.***

Method 3

- An alternate way is that a task-instruction is executed on the same processor or different instructions of a task can be done on different processors (loosely coupled).
- A compiler schedules the various instructions of the tasks among the processors at an instance

Performance cost

- Suppose one processor finishes computations earlier than the other.
- Performance cost is more if there is idle time left from the available. time.

Performance cost

- If one task needs to send a message to another and the other waits (blocks) till the message is received, performance cost is proportional to the waiting period.

Summary

We learnt

- A multiprocessor system uses two or more processors for faster execution of the (i) Program functions, (ii) tasks or (iii) single instruction multiple data instructions
- Partitioning of the processes and scheduling of processes on multiple processor can be static or dynamic

End of Lesson 7 of Chapter 6