MULTIPROGRAMMING AND MULTITHREADING

	3:26 / 3:26	1.0x			

Download transcript .txt

Show Discussion New Post

THREAD-LEVEL PARALLELISM

Help

	4:10 / 4:10	1.0x			

Download transcript .txt

Show Discussion New Post

1. CHECK YOUR UNDERSTANDING (1/1 point)

In the video, the two-way multithreaded pipeline executed the following two threads:

```
Thread 1                    Thread 2

LOOP  ADD   R1, R2, R3      ADD   R7, R1, 0
      NOT   R4, R1          ADD   R4, R7, R2
      AND   R5, R4, R3      AND   R5, R4, 1
      ADD   R6, R5, 2       STR   R5, R1, 0
      BRz   LOOP            ADD   R1, R1, 0
```

How was the two-way multithreaded pipeline able to achieve higher throughput than the 2-way superscalar pipeline without multithreading?

- Data dependences were completely removed from both programs.
- The hardware was able to pair one instruction from each thread to move through the pipeline together. ✔
- The compiler was able to reorder instructions to address data dependences.
- The two-way multithreaded pipeline did not require any forwarding of data.

Help

EXPLANATION

The first answer is incorrect since the data dependences are still present in both threads.

The second answer is true. As we saw in the video, the hardware moved pairs of instructions, one from each thread, through the pipeline.

The compiler did not reorder instructions.

Forwarding was in fact performed within the pipeline. For instance, when the first instruction in each thread (the two ADDs) moved together into MEM, their results were forwarded into EX for the following NOT and ADD instructions.

Final Check | **Save** | Hide Answer *You have used 1 of 2 submissions*

Show Discussion

New Post