

Ch 4.4 More microarchitecture

Ch 4.4 deepens our understanding of microarchitecture design, focusing primarily on speeding up our Mic-1 micro-architecture.

The greatest speedups come from hardware advances : Moore's Law.

We can also speedup our microarchitecture in 3 ways:

1. Reduce the num clock cycles needed per IJVM instruction
2. Simplify organization so that clock cycle is shorter => faster clock speed
3. Overlap the execution of instructions => pipelining

Mic-2 speedup

There are three speedups in Mic-2:

- Eliminating Main1 - goto the next opcode directly with goto (MBR)
- Three bus architecture - Separate A bus, B bus feed into ALU
- Add an Instruction Fetch Unit (IFU) - don't use the ALU to increment PC

Mic-3 speedup

The Mic-3 microarchitecture introduces a 4 stage pipeline. We'll insert latches for A, B, and C busses so that we can slice datapath execution into 3 micro-steps:

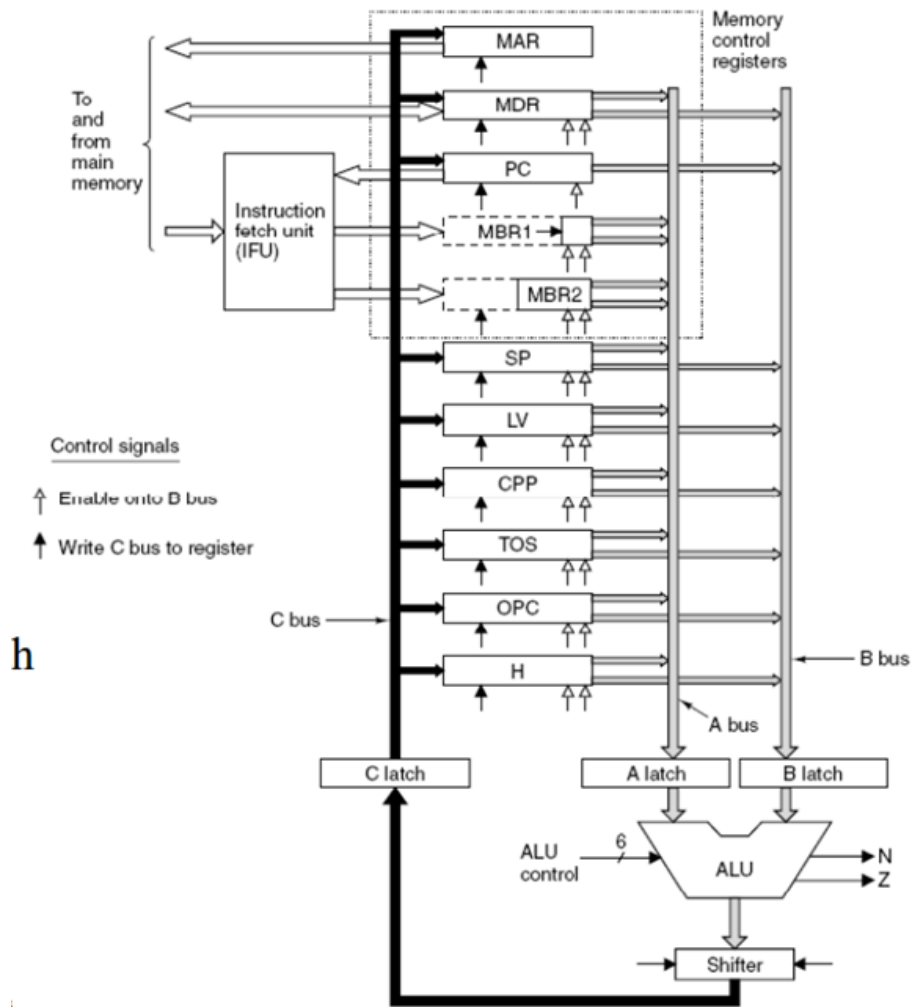
- Load A and B busses from registers
- Perform ALU, shift operations
- Write registers from C bus

Two benefits:

1. Each micro-step is faster, so we can increase clock speed
2. Each micro-step is isolated (clocked by latches and registers), so we can execute them independently and therefore pipeline them

Mic-3 changes from Mic-1 are:

- Add the A bus (Mic-2 change)... to reduce micro-instruction
- Add the Instruction Fetch Unit (Mic-2 change)... so ALU is not used for PC++
- Add the bus latches... to facilitate pipelining



Pipelining example: The SWAP instruction

	Swap1	Swap2	Swap3	Swap4	Swap5	Swap6
Cy	MAR=SP-1;rd	MAR=SP	H=MDR;wr	MDR=TOS	MAR=SP-1;wr	TOS=H;goto (MBR1)
1	B=SP					
2	C=B-1	B=SP				
3	MAR=C; rd	C=B				
4	MDR=Mem	MAR=C				
5			B=MDR			
6			C=B	B=TOS		
7			H=C; wr	C=B	B=SP	
8			Mem=MDR	MDR=C	C=B-1	B=H
9					MAR=C; wr	C=B
10					Mem=MDR	TOS=C
11						goto (MBR1)

Figure 4-33. The implementation of SWAP on the Mic-3.

Mic-4 speedup

The Mic-4 micro-architecture introduces a 7-stage pipeline.