Advanced Computer Architectures

- **Memorandum Protocol**
- Portfolio Exam WiSe 20223 (in present)
- 50 points in total, 75min

1. Performance (10P)

Type	Percentage	Cycles
ALU	50%	1
Floating Points	30%	3
Load / Store	10%	4
Branch / Jump	10%	2

- a) Calculate the average CPI
- b) We improve the hardware. Now Branch and Jump only take 1 cycle. What is the overall performance improvement ?
- c) Now we are only improving Floating Points. The minimum is 1 cycle. Can we reach an overall performance improvement of 50% ? What is the maximum possible performance improvement ?

2.) Dependencies and pipelined Execution (12P)

- The following Assmbly Code is given:

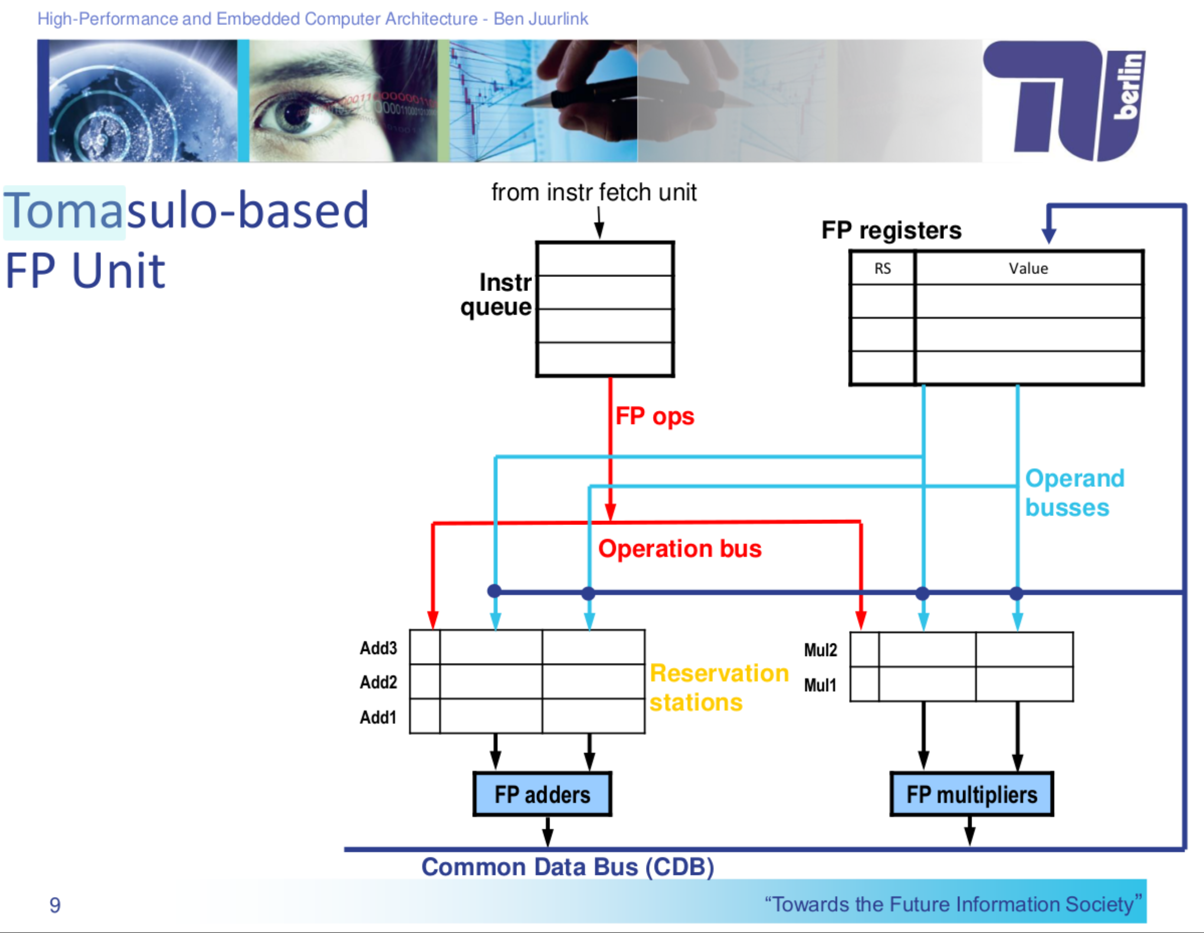
```
ADD R2,R3,R4
SLL R5,R2,#2
LW R6,12(R5)
SUB R8,R6,R3
BNE R8,R1,L2
ADDI R3,R3,#1
L2: ADD R2,R3,R4
```

- a) Identify all controll and true data dependencies. Draw arrow between two instructions if the second instructions depends on the first instruction and name the type of dependency. You do not need to draw transistive dependencies
- b) Given a 5 stage canoical pipeline with hardware forwarding. The branch condition and target is calculated in ID. Complete the table. Draw an arrow when forwarding is needed and an X if the processor stalls.

Instruction	cc1	cc2	cc3	cc4	cc5	cc6	cc7	cc8	cc9	cc10	cc11	cc12	c
ADD R2,R3,R4													
SLL R5,R2,#2													
LW R6,12(R5)													
SUB R8,R6,R3													
BNE R8,R1,L2													
ADDI R3,R3,#1													
L2: ADD R2,R3,R4													

3.) Tomasulo's Algorithm (6P)

- Graphic given describing Tomasulo's Algorihm (Example from the slides)


High-Performance and Embedded Computer Architecture - Ben Juurlink

Tomasulo-based FP Unit

9 “Towards the Future Information Society”

- Instruction queue:

	Instruction Queue
	MUL.D F3, F3, F0
	ADD.D F0, F3, F1

- Reservation Station Adder

RS	Op	Val1	Val2
Add3			
Add2			
Add1	+	Mul1	2.0
RS	Op	Val1	Val2

- Reservation Station Multiplier

RS	Op	Val1	Val2
Mul2			
Mul1	*	1.0	2.0

- Show the content of all entities after the two instructions are queued. You may need to replace content

4.) Superscalar (12P)

- dual issue processor
- 1 integer FU (ALU + EA)
- seperate FU for branches
- seperate pipelined FU for FP adder
- 2 seperate pipelined FU for FP multiplier
- issue/write to CDB 1cycle
- commit width = issue width
- perfect branch prediction with speculation
- latencies:
  - 1cc int ALU
  - 2cc load / store
  - 3cc FP adder
  - 4cc FP multiplier
- Write which instruction is in which cycle in which stage

Iteration	Instruction	Issue	Execute	Memory	CDB	Commit	Comment
1	L.D. F3,0(R1)						
1	MUL.D F5,F3,F3						
1	MUL.D F6,F3,F0						
1	ADD.D F7,F5,F6						
1	S.D F8,0(R1)						
1	DADDI R1,R1,#8						
1	BNE R1,R5,L1						
2	L.D. F3,0(R1)						
2	MUL.D F5,F3,F3						
2	MUL.D F6,F3,F0						
2	ADD.D F7,F5,F6						
2	S.D F8,0(R1)						
2	DADDI R1,R1,#8						
2	BNE R1,R5,L1						

5.) Cache (4P)

- L1 direct-mapped cache given with 16kb (1k = 1024) and a block size of 64b
- Caculcate the **index** for the following addresses and state whether it yield a **hit or miss** if the cache starts empty
  - 16384
  - 16408
  - 2048
  - 2064

6.) Branch Prediction (6P)

- a) Explain the following branch predictors
  - 1bit branch prediction (branch history table)

  - 2bit branch prediction (bimodal branch prediction)
- b) Draw the state machine of an 2bit branch predictor
- c) Branch has the following behaviour. Assuming the initial prediction is "weakly NT" 01, how often would 2-bit BHT (bimodal predictor) misspredicts ?
  - T, T, NT, NT, NT, T, T, NT