Aufgabe 1

a)

Instruction Memory:	250ps
PC-Adder:	125ps
Register file lesen	100ps
2xPre-ALU-MUX:	60ps
Control Unit:	75ps
ALU:	200ps
Data Memory:	275ps
Post-ALU-MUX:	30ps
Register file schreiben:	125ps
2x pre PC Mux:	60ps
Summe:	1300ps

b)

Fetch:

PC-Adder:	125ps
Instruction Memory:	250ps
2x pre PC Mux:	60ps
Summe	335ps

Decode:

Control Unit:	75ps
Register lesen:	100ps
1xPre ALU Mux:	30ps
Summe	205ps

Execute:

1xPre ALU Mux:	30ps
ALU:	200ps
Summe	230ps

Memory:

Data Memory:	275ps
Summe:	275ps

Write Back:

Post ALU Mux:	30ps
Register File schreiben:	125ps
Summe:	155ps

Minimale Länge: 335ps

Theoretisch kann ein Speedup mit dem Faktor 5 erreicht werden, da eine 5 stufige Pipeline verwendet wird.

Vergleich mit komb. aus Übung 5:

905 / 335 = 2,701 => Speedup von Faktor 2,701

Nicht jede Stufe dauert gleich lange, daher ist der Speedup nicht gleich 5.

Aufgabe 2

sub	$t0	$t1	$t2
addi	$t2	$t3	-2
add	$t4	$t2	$t1
beq	$t0	$0	else
addi	$t2	$t5	9
lw	$t0	8($t3)	
sub	$t3	$t0	$t5
j	end		

else:

sw	$t3	4($t0)	
or	$t0	$t3	$t4
add	$t5	$t2	$t4
xor	$t4	$t1	$t0

end:

xor	$t2	$t3	$t0
and	$t6	$t6	$t1

a)
RAW zwischen S2 und S3 mit t2
RAW zwischen S1 und S4 mit t0
WAW zwischen S2 und S5 mit t2
WAW zwischen S7 und S1 mit t0
RAW zwischen S9 und S7 mit t3
WAW zwischen S7 und S9 mit t0
WAR zsichen S10, S11 und S12 mit t4
RAW zwischen S9 und S12 mit t0
WAR zwischen S13 und S11 mit t2

b)

S1 reads t1, t2 writes t0
S2 reads t3 writes t2
S3 reads t2, t1 writes t4
S4 reads t0 0
S5 reads t5 writes t2
S6 reads t3 writes t0
S7 reads t0, t5 writes t3
S8 jumps to end
~~else:~~
S9 reads t3 writes t0
~~S10 reads t3, t4 writes t0~~
~~S11 reads t2, t4 writes t0~~
~~S12 reads t1, t0 writes t4~~
end:
S13 reads t3, t0 writes t2
S14 reads t3, t0 writes t6

Timestamp	Fetch	Decode	Execute	Memory Access	Write Back
1	sub				
2	addi	sub			
3	add	addi	sub		
4	beq	add (1)	addi (1)	sub	
5	addi	beq (2)	add	addi	sub (2)
6	lw (1)	addi (1)	beq (1)	add	addi
7	sub	lw	addi	beq	add
8	j	sub (3)	lw (3)	addi	beq

9	sw (1)	j (1)	sub	lw	addi
10	xor	sw (4)	j	sub (4)	lw
11	and	xor	sw	j	sub
12		and	xor	sw	j
13			and	xor	sw
14				and	xor
15					and

Data Hazards:

1. In step 4, add reads t2, but addi has not yet written its value back into memory
2. In step 5, beq reads from t0, but sub has not yet written its value back into memory
3. In step 8, sub reads from t0, but lw has not yet written its value back into memory
4. In step 10, sw reads from t3, but sub has not yet written its value back into memory

Control Hazards:

1. In step 9, j is executed in the decode phase, but the execution of sw has already started

S1 reads t1, t2 writes t0
S2 reads t3 writes t2
S3 reads t2, t1 writes t4
S4 reads t0 0
S5 reads t5 writes t2
S6 reads t3 writes t0
~~S7 reads t0, t5 writes t3~~
~~S8 jumps to end~~
else:
S9 reads t3 writes t0
S10 reads t3, t4 writes t0
S11 reads t2, t4 writes t5
S12 reads t1, t0 writes t4
end:
S13 reads t3, t0 writes t2
S14 reads t3, t0 writes t6

Timestamp	Fetch	Decode	Execute	Memory Access	Write Back
1	sub				
2	addi	sub			
3	add	addi	sub		
4	beq	add (1)	addi (1)	sub	
5	addi	beq (2)	add	addi	sub (2)
6	lw (1)	addi (1)	beq (1)	add	addi
7	sw	lw	addi	beq	add
8	or	sw	lw	addi	beq
9	add	or	sw	lw	addi
10	xor	add	or	sw	lw
11	xor	xor	add	or	sw
12	and	xor	xor	add	or
13		and	xor	xor	add
14			and	xor	xor
15				and	xor
16					and

Data Hazards:

1. In step 4, add reads t2, but addi has not yet written its value back into memory
2. In step 5, beq reads from t0, but sub has not yet written its value back into memory

Control Hazards:

1. In step 6, beq is being executed, but the execution of lw and addi has already started

c)

S1	reads t1, t2	writes t0
S2	reads t3	writes t2
S3	reads t2, t1	writes t4
S4	reads t0 0	
S5	reads t5	writes t2
S6	reads t3	writes t0
S7	reads t0, t5	writes t3
S8	jumps to end	
~~else:~~		
S9	reads t3	writes t0
~~S10~~	~~reads t3, t4~~	~~writes t0~~
~~S11~~	~~reads t2, t4~~	~~writes t0~~
~~S12~~	~~reads t1, t0~~	~~writes t4~~
end:		
S13	reads t3, t0	writes t2
S14	reads t3, t0	writes t6

Timestamp	Fetch	Decode	Execute	Memory Access	Write Back
1	sub				
2	addi	sub			
3	nop	addi	sub		
4	nop	nop	addi	sub	
5	add	nop	nop	addi	sub
6	beq	add	nop	nop	addi
7	lw	beq	add	nop	nop
8	nop	lw	beq	add	nop
9	nop	nop	lw	beq	add
10	sub	nop	nop	lw	beq
11	j	sub	nop	nop	lw
12	nop	j	sub	nop	nop
13	xor	nop	j	sub	nop
14	and	xor	nop	j	sub
15		and	xor	nop	j

16			and	xor	nop
17				and	xor
18					and

S1 reads t1, t2 writes t0
S2 reads t3 writes t2
S3 reads t2, t1 writes t4
S4 reads t0 0
S5 reads t5 writes t2
S6 reads t3 writes t0
~~S7 reads t0, t5 writes t3~~
~~S8 jumps to end~~
else:
S9 reads t3 writes t0
S10 reads t3, t4 writes t0
S11 reads t2, t4 writes t5
S12 reads t1, t0 writes t4
end:
S13 reads t3, t0 writes t2
S14 reads t3, t0 writes t6

Timestamp	Fetch	Decode	Execute	Memory Access	Write Back
1	sub				
2	nop	sub			
3	nop	nop	sub		
4	addi	nop	nop	sub	
5	beq	addi	nop	nop	sub
6	nop	beq	addi	nop	nop
7	nop	nop	beq	addi	nop
8	sw	nop	nop	beq	addi
9	or	sw	nop	nop	beq
10	add	or	sw	nop	nop
11	xor	add	or	sw	nop

12	xor	xor	add	or	sw
13	and	xor	xor	add	or
14		and	xor	xor	add
15			and	xor	xor
16				and	xor
17					and

d)

S1	reads t1, t2	writes t0
S2	reads t3	writes t2
S3	reads t2, t1	writes t4
S4	reads t0 0	
S5	reads t5	writes t2
S6	reads t3	writes t0
S7	reads t0, t5	writes t3
S8	jumps to end	
~~else:~~		
S9	reads t3	writes t0
~~S10~~	~~reads t3, t4~~	~~writes t0~~
~~S11~~	~~reads t2, t4~~	~~writes t0~~
~~S12~~	~~reads t1, t0~~	~~writes t4~~
end:		
S13	reads t3, t0	writes t2
S14	reads t3, t0	writes t6

Timestamp	Fetch	Decode	Execute	Memory Access	Write Back
1	sub				
2	addi	sub			
3	nop	addi	sub		
4	nop	nop	addi	sub	
5	add	nop	nop	addi	sub
6	beq	add	nop	nop	addi
7	lw	beq	add	nop	nop
8	nop	lw	beq	add	nop
9	nop	nop	lw	beq	add
10	sub	nop	nop	lw	beq
11	j	sub	nop	nop	lw
12	xor	j	sub	nop	nop
13	and	xor	j	sub	nop
14		and	xor	j	sub
15			and	xor	j

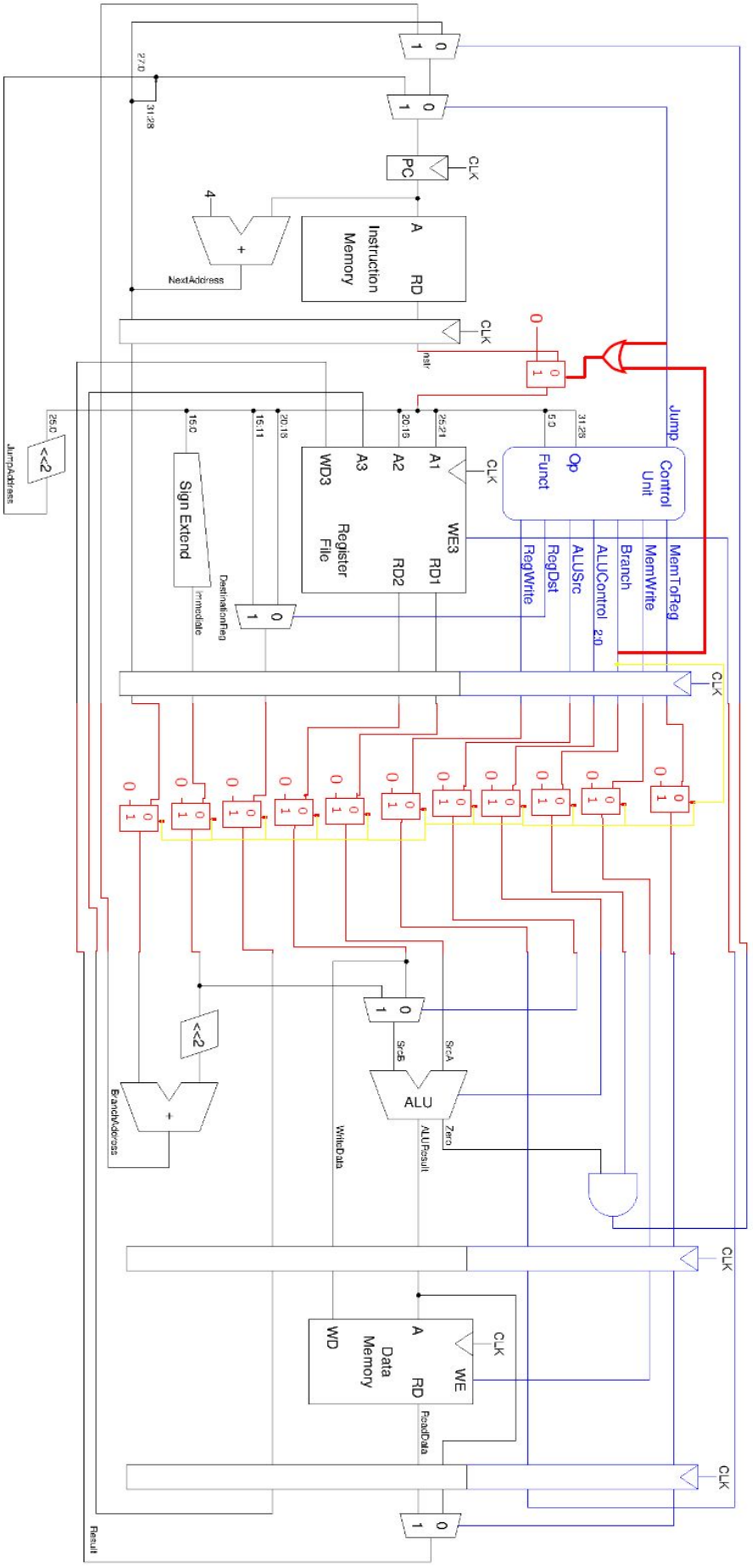
16				and	xor
17					and
18					

S1 reads t1, t2 writes t0
S2 reads t3 writes t2
S3 reads t2, t1 writes t4
S4 reads t0 0
S5 reads t5 writes t2
S6 reads t3 writes t0
~~S7 reads t0, t5 writes t3~~
~~S8 jumps to end~~
else:
S9 reads t3 writes t0
S10 reads t3, t4 writes t0
S11 reads t2, t4 writes t5
S12 reads t1, t0 writes t4
end:
S13 reads t3, t0 writes t2
S14 reads t3, t0 writes t6

Timestamp	Fetch	Decode	Execute	Memory Access	Write Back
1	sub				
2	nop	sub			
3	nop	nop	sub		
4	addi	nop	nop	sub	
5	beq	addi	nop	nop	sub
6	sw	beq	addi	nop	nop
7	or	sw	beq	addi	nop
8	add	or	sw	beq	addi
9	xor	add	or	sw	beq
10	xor	xor	add	or	sw
11	and	xor	xor	add	or

12		and	xor	xor	add
13			and	xor	xor
14				and	xor
15					and

Aufgabe 3


PC
CLK
4
+
NextAddress
Instruction Memory
A
RD
Instr
Jump
Control Unit
Op
Funct
MemToReg
MemWrite
Branch
ALUControl 2:0
ALUSrc
RegDst
RegWrite
Register File
WD3
A3
A2
A1
WE3
RD1
RD2
Sign Extend
Immediate
DestinationReg
JumpAddress
<<2
BranchAddress
SrcB
SrcA
ALU
Zero
ALUResult
WriteData
Data Memory
WD
WE
ReadData
Result