

# Advanced and parallel architectures

Prof. A. Massini

Midterm Exam – April 20, 2017

- Student's Name -

---

- *Matricola* number -

---

Exercise 1 (3 points)	
Exercise 2 (3 points)	
Exercise 3a (3 points)	
Exercise 3b (3 points)	
Exercise 4 (4 points)	
Exercise 5 (4 points)	
Exercise 6a (2 points)	
Exercise 6b (3 points)	
Exercise 7a (2 points)	
Exercise 7b (2 points)	
Exercise 8 (3 points)	
Total (32 points)	

### Exercise 1 (3 points) – Instruction pipeline

Consider the following loop expressed in a high level language:

for (i =0; i < N; i ++)

```
{
    vectA[i] = vectA[i] + k;
    vectC[i] = vectB[i] + vectC[i];
}
```

The program has been written in MIPS assembly code, assuming that registers \$t6 and \$t7 have been initialized with values 0 and 4N respectively. The symbols VECTA, VECTB and VECTC is a 16-bit constant.

Let us consider the loop executed by 5-stage pipelined MIPS processor WITHOUT any optimisation in the pipeline.

1. Identify the Hazard Type (Data Hazard or Control Hazard) in the last column
2. In the first column identify the number of stalls to be inserted before each instruction (or between stages IF and ID of each instruction) necessary to solve the hazards
3. For each hazard, add an ARROW to indicate the pipeline stages involved in the hazard

Num. Stalls	INSTRUCTION	C1	C2	C3	C4	C5	C7	C6	C8	C9	C10	C11	C12	C13	C14	Hazard Type
	FOR: beq \$t6,\$t7,END	IF	ID	EX	ME	WB										
	lw \$t2,VECTA(\$t6)		IF	ID	EX	ME	WB									
	lw \$t3,VECTB(\$t6)			IF	ID	EX	ME	WB								
	lw \$t4,VECTC(\$t6)				IF	ID	EX	ME	WB							
	addi \$t2,\$t2,k					IF	ID	EX	ME	WB						
	sw \$t2,VECTA(\$t6)						IF	ID	EX	ME	WB					
	add \$t4,\$t3,\$t4							IF	ID	EX	ME	WB				
	sw \$t4,VECTC(\$t6)								IF	ID	EX	ME	WB			
	addi \$t6,\$t6,4									IF	ID	EX	ME	WB		
	j FOR										IF	ID	EX	ME	WB	



### Exercise 3a (3 points) - Number representation

Represent the natural number range  $[0; 255]$  using the **residue number system**, considering different choices of the moduli set (at least three different choices). Compare the different choices with respect to the number of bits necessary for the representation. Consider also the number of bits needed for representing the range  $[0; 255]$  with the conventional binary system.

Represent  $A=34$  and  $B=21$  using the considered different choices of the moduli sets, and show how to compute the sum  $A+B$ .

**Exercise 3b (3 points) - Circuit time and area**

Compute the time (propagation delay) and area required by the adders needed for the moduli sets considered in the previous point, using the appropriate ripple-carry adder for each modulo.

Compute the speed-up of each choice with respect to the standard binary ripple-carry adder, that is the ratio:  $S = \text{residue number system time} / \text{binary number system time}$ .

**Exercise 4 (4 points) Pipelined operations**

Given the values  $A=7$  and  $B=3$ , show the scheme and the execution of the pipelined multiplications:  $A \times B$  and  $A \times (-B)$ . Verify the results.

**Exercise 5 (4 points) Number representation**

Given the value  $Y = \bar{1}01\bar{1}01$  expressed in the MSD number system, give the correspondent value in the decimal system.

---

Give at least three MSD expressions of  $X= 37$ .

---

Show how to execute  $X+Y$  using the look-up tables shown below here.

		First addend		
		-1	0	1
Second addend	-1	0	1	0
		-1	-1	0
	0	1	0	-1
		-1	0	1
1	0	-1	0	
		0	1	1

		First addend		
		-1	0	1
Second addend	-1	0	-1	0
		-1	0	0
	0	-1	0	1
		0	0	0
1	0	1	0	
		0	1	

**Exercise 6a (2 points) – Number representation**

Given the values  $A = 00\ 11\ 10\ 01\ 11$  and  $B = 00\ 10\ 10\ 11\ 01$  in the signed RB (Redundant Binary) representation, convert A and B in decimal.

**Exercise 6b (3 points) – Number representation**

Show the execution of operation  $A - B$ . Verify the value of the results.



### Exercise 7a (2 points) – Loop dependences

In the following loop, find all the true dependences, output dependences and antidependences.

Eliminate the output dependences and antidependences by renaming.

```
for (i=1; i<=n; i++) {  
    Z[i]=X[i]+2;      \*S1*\  
    W[i]=Y[i]-1;     \*S2*\  
    X[i]=U[i]+V[i];  \*S3*\  
    X[i]=Z[i]+3;     \*S4*\  
    Y[i]=X[i]*2;     \*S5*\  
}
```

### Exercise 7b (2 points) – Loop dependences

In the following loops, find the loop carried dependences with respect to index i and/or index j. If possible, propose a solution to parallelize the loops.

```
for (i=1; i<n; i++) {  
    a[i] = 2*a[i-1]; \*S1*\  
    b[i] = b[i-1]/2; \*S2*\  
    c[i] = a[i]+b[i]; \*S3*\  
}
```

```
for (i=1; i<n; i++) {  
    for (j=1; j<n; j++)  
        v[i][j] = v[i][j-1] + 4*j; \*T1*\  
        w[i][j] = w[i-1][j] - 2*i; \*T2*\  
}
```

**Exercise 8 (3 points) – Arithmetic operations**

Show the execution of the multiplication  $A \times B$ , where  $A=10010101$  and  $B=001111110$ , in the standard way and using the Booth recoding. Explain what is the gain when Booth recoding is applied.