

Ejemplos-codigo-examen-tema-4-re...



KIKONASO



Arquitectura de Computadores



2º Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada**



Estamos de
Aniversario

De la universidad al
mercado laboral:
especialízate con los posgrados
de EOI y marca la diferencia.



EOI Escuela de
organización
industrial



saber más

		instrucción	operandos	explicación	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	ADD/SUB	1	xorq	%rbx, %rbx	; rbx <- rbx xor rbx	1F	1D	EX														
	ADD/SUB	2	addq	\$N, %rax	; rax <- rax + N	1F	1D	EX														
Etapas	LOAD	3	movsd	x(%rax), %xmm1	; xmm1 <- M[x+rax]	1F	1D		EX	EX												
cauce	MUL	4	mulsd	%xmm1, %xmm1	; xmm1 <- xmm1 * xmm1	1F	1D			EX	EX	EX	EX									
Se procesan		5	movsd	y(%rax), %xmm0	; xmm0 <- M[y+rax]		1F	1D						EX	EX							
1 vez	MUL	6	mulsd	%xmm0, %xmm0	; xmm0 <- xmm0 * xmm0		1F	1D								EX	EX	EX	EX			
las instruc.		7	addsd	%xmm1, %xmm0	; xmm0 <- xmm0 + xmm1		1F	1D												EX		
1 a 12	LD	8	movsd	sq(%rbx), %xmm3	; xmm3 <- M[sq+rbx]		1F	1D												EX		
	ADD/SUB	9	addsd	%xmm0, %xmm3	; xmm3 <- xmm3 + xmm0			1F	1D												EX	
	STORE	10	movsd	%xmm3, sq(%rbx)	; M[sq+rbx] <- xmm3			1F	1D												EX	
	ADD/SUB	11	dec	%ecx	; ecx <- ecx - 1			1F	1D												EX	
	ADD/SUB	12	subq	\$8, %rax	; rax <- rax - 8			1F	1D												EX	
bits válido		1	xorq	%rbx, %rbx	; rbx <- rbx xor rbx	11	11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ventana(s)		2	addq	\$N, %rax	; rax <- rax + N	11	11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
instrucciones		3	movsd	x(%rax), %xmm1	; xmm1 <- M[x+rax]	0	0	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-
o		4	mulsd	%xmm1, %xmm1	; xmm1 <- xmm1 * xmm1	00	09	-	-	-	11	-	-	-	-	-	-	-	-	-	-	-
estación o		5	movsd	y(%rax), %xmm0	; xmm0 <- M[y+rax]	0	0	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-
estaciones		6	mulsd	%xmm0, %xmm0	; xmm0 <- xmm0 * xmm0	00	09	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
de reserva		7	addsd	%xmm1, %xmm0	; xmm0 <- xmm0 + xmm1	00	09	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(al finalizar		8	movsd	sq(%rbx), %xmm3	; xmm3 <- M[sq+rbx]	0	0	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-
el ciclo)		9	addsd	%xmm0, %xmm3	; xmm3 <- xmm3 + xmm0	00	09	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		10	movsd	%xmm3, sq(%rbx)	; M[sq+rbx] <- xmm3	00	09	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		11	dec	%ecx	; ecx <- ecx - 1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		12	subq	\$8, %rax	; rax <- rax - 8	10	10	-	11	-	-	-	-	-	-	-	-	-	-	-	-	-

ESPECIFICACIONES:

1/2 UF PARA CARGA (2C)

1/2 UF PARA ALMACENAMIENTO (1C)

3 UF ADD/SUB (3C)

1 UF MUL (4C)

VENTANA CENTRALIZADA Y EMISIÓN ORDENADA

CAPTAR, DECODIFICAR Y EMITIR 4 INS/CICLO

verano
empleo.

		instrucción	operandos	explicación	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Etapas cauce Se procesan 1 vez las instruc. 1 a 12	1	xorg	%rbx, %rbx	: rbx <- rbx xor rbx	IF	ID	EX													
	2	addq	\$N, %rax	: rax <- rax + N	IF	ID	EX													
	3	movsd	x(%rax), %xmm1	: xmm1 <- M[x+rax]	IF	ID	EX													
	4	mulsd	%xmm1, %xmm1	: xmm1 <- xmm1 * xmm1	IF	ID		EX	EX											
	5	movsd	y(%rax), %xmm0	: xmm0 <- M[y+rax]		IF	ID			EX	EX									
	6	mulsd	%xmm0, %xmm0	: xmm0 <- xmm0 * xmm0		IF	ID				EX	EX								
	7	addsd	%xmm1, %xmm0	: xmm0 <- xmm0 + xmm1		IF	ID						EX	EX	EX	EX				
	8	movsd	sq(%rbx), %xmm3	: xmm3 <- M[sq+rbx]		IF	ID						EX	EX				EX		
	9	addsd	%xmm0, %xmm3	: xmm3 <- xmm3 + xmm0			ID	ID											EX	
	10	movsd	%xmm3, sq(%rbx)	: M[sq+rbx] <- xmm3			IF	ID												EX
	11	dec	%ecx	: ecx <- ecx - 1			IF	ID	EX											
	12	subq	\$8, %rax	: rax <- rax - 8			IF	ID	EX											
bits válido	1	xorg	%rbx, %rbx	: rbx <- rbx xor rbx	11	11	11	11	11	11	11	-	-	-	-	-	-	-	-	-
ventana(s)	2	addq	\$N, %rax	: rax <- rax + N	11	11	11	11	11	11	11	-	-	-	-	-	-	-	-	-
instrucciones	3	movsd	x(%rax), %xmm1	: xmm1 <- M[x+rax]	0	0	1	1	1	1	1	-	-	-	-	-	-	-	-	-
o	4	mulsd	%xmm1, %xmm1	: xmm1 <- xmm1 * xmm1	00	00	00	00	00	00	00	-	-	-	-	-	-	-	-	-
estación o	5	movsd	y(%rax), %xmm0	: xmm0 <- M[y+rax]	0	0	1	1	1	1	1	-	-	-	-	-	-	-	-	-
estaciones	6	mulsd	%xmm0, %xmm0	: xmm0 <- xmm0 * xmm0	00	00	00	00	00	00	00	11	11	-	-	-	-	-	-	-
de reserva	7	addsd	%xmm1, %xmm0	: xmm0 <- xmm0 + xmm1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
(al finalizar	8	movsd	sq(%rbx), %xmm3	: xmm3 <- M[sq+rbx]	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
el ciclo)	9	addsd	%xmm0, %xmm3	: xmm3 <- xmm3 + xmm0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	10	movsd	%xmm3, sq(%rbx)	: M[sq+rbx] <- xmm3	00	00	10	10	10	10	10	10	10	10	10	10	10	10	10	10
	11	dec	%ecx	: ecx <- ecx - 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	12	subq	\$8, %rax	: rax <- rax - 8	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

3 UF PARA CARGA (ZC)

1 UF PARA STORE (ZC)

3 UF ADD/SUB (ZC)

3 UF MUL (ZC)

VENTANA CENTRALIZADA Y EMISIÓN DESORDENADA

CAPTAR, DECODIFICAR Y EMITIR 4 MB/s

		instrucción	operandos	explicación	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	1	xorq	%rbx, %rbx	; rbx <- rbx xor rbx	IF	ID	EX	WB												
	2	addq	\$N, %rax	; rax <- rax + N	IF	ID	EX	WB												
Etapas	3	movsd	x(%rax), %xmm1	; xmm1 <- M[x+rax]	ID	ID	EX	WB												
cauce	4	mulsd	%xmm1, %xmm1	; xmm1 <- xmm1 * xmm1	IF	ID		EX	EX	WB										
Se procesan	5	movsd	y(%rax), %xmm0	; xmm0 <- M[y+rax]	IF	ID	EX	EX		EX	EX	EX	EX	WB						
1 vez	6	mulsd	%xmm0, %xmm0	; xmm0 <- xmm0 * xmm0		IF	ID			EX	EX	EX	EX	WB						
las instruc.	7	addsd	%xmm1, %xmm0	; xmm0 <- xmm0 + xmm1		IF	ID			EX	EX	EX	EX	WB						
1 a 12	8	movsd	sq(%rbx), %xmm3	; xmm3 <- M[sq+rbx]		IF	ID			EX	EX			EX	WB					
	9	addsd	%xmm0, %xmm3	; xmm3 <- xmm3 + xmm0			IF	ID						EX	WB					
	10	movsd	%xmm3, sq(%rbx)	; M[sq+rbx] <- xmm3			IF	ID							EX	WB				
	11	dec	%ecx	; ecx <- ecx - 1			IF	ID	EX											
	12	subq	\$8, %rax	; rax <- rax - 8			ID	ID	EX											
bits válido	1	xorq	%rbx, %rbx	; rbx <- rbx xor rbx	11	11	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ventana(s)	2	addq	\$N, %rax	; rax <- rax + N	11	11	-	-	-	-	-	-	-	-	-	-	-	-	-	-
instrucciones	3	movsd	x(%rax), %xmm1	; xmm1 <- M[x+rax]	0	0	0	1	1	-	-	-	-	-	-	-	-	-	-	-
o	4	mulsd	%xmm1, %xmm1	; xmm1 <- xmm1 * xmm1	00	00	00	00	00	11	-	-	-	-	-	-	-	-	-	-
estación o	5	movsd	y(%rax), %xmm0	; xmm0 <- M[y+rax]	0	0	0	1	-	-	-	-	-	-	-	-	-	-	-	-
estaciones	6	mulsd	%xmm0, %xmm0	; xmm0 <- xmm0 * xmm0	00	00	00	00	00	11	-	-	-	-	-	-	-	-	-	-
de reserva	7	addsd	%xmm1, %xmm0	; xmm0 <- xmm0 + xmm1	00	00	00	00	00	00	00	00	00	11	-	-	-	-	-	-
(al finalizar	8	movsd	sq(%rbx), %xmm3	; xmm3 <- M[sq+rbx]	0	0	0	1	1	-	-	-	-	-	-	-	-	-	-	-
el ciclo)	9	addsd	%xmm0, %xmm3	; xmm3 <- xmm3 + xmm0	00	00	00	00	00	00	00	10	10	10	11	-	-	-	-	-
	10	movsd	%xmm3, sq(%rbx)	; M[sq+rbx] <- xmm3	00	00	00	00	10	10	10	10	10	10	10	11	-	-	-	-
	11	dec	%ecx	; ecx <- ecx - 1	1	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-
	12	subq	\$8, %rax	; rax <- rax - 8	10	10	10	11	11	-	-	-	-	-	-	-	-	-	-	-

ESPECIFICACIONES:

- 2 UF CARGA (7c)
- 2 UF ADD/SUB (3c)
- 2 UF MUL (4c)
- 1 UF ALMACENAMIENTO (3c)

VENTANA CENTRALIZADA Y EMISIÓN DESORDENADA
CAPTAR, DECODIFICAR Y EMITIR M INSTRUCCIONES