

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



WUOLAH

		instrucción	operandos	explicación	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
ADDIGUD	1	xorq	%rbx, %rbx	; rhx <- rbx xor rbx	15	10	EX														-
ADD/SUD	2	addq	\$N, %rax	: rax <- rax + N	1+	15	EX												1-20		
Etapas LOAD	3	movsd	x(%rax), %xmml	: xmml <- M[x+rax]	15	10		EX	X												
cauce Mul	4	mulsd	%xmml, %xmml	; xmm1 <- xmm1 * xmm1	1=	10				EX	EX	EX	EX		1000					1 30	
Se procesanil	5	movsd	y(%rax), %xmm0	; xmm0 <- M[y+rax]		11	10							Ex	ΈX						
1 vez Mul	6	mulsd	%xmm0, %xmm0	; xmm0 <- xmm0 * xmm0		IF	10									tex	23	EX	EX		
Blas instruc.	7	addsd	%xmml, %xmm0	; xmm0 <- xmm0 + xmm1		IF	10													EX	
1 a 12 11		movsd	sq(%rbx), %xmm3	; xmm3 <- M[sq+rbx]		14	10													FX	
ADDRUB	9	addsd	%xmm0, %xmm3	; xmm3 <- xmm3 + xmm0			1F	10													10
STOOL	10	movsd	%xmm3, sq(%rbx)	; M[sq+rbx] <- xmm3			11	115													
ADD/SUB	11	dec	%ecx	; ecx <- ecx - 1			IP	ID													
	12	subq	\$8, %rax	; rax <- <u>rax</u> - 8			11	10			1577										
	7	vora	&rhy &rhy	: rbx <- rbx xor rbx	11	11	-	-	-	-	-	-	· ·	-	-	-	-	-	-		
	_	-			1000		-	-	-	-	-	-	-	-	-	-	-	-		-	
ALL DESCRIPTION OF THE PARTY OF	_				-	0	-	1	-	-	-	-	-	-	-	-	-	-		-	
					00	09	-	-	-	11	-	-	-	-	-	-	-	-	-		
	=			; xmm0 <- M[y+rax]	(0 0	-	1	-	-	-	-	-	-	-	-	-	-		-	
	_			; xmm0 <- xmm0 * xmm0	00	00	-	-	-	-	-	-	-	-	-	-	-	-			
	=					00	-	-	-	-	-	-	-	-	-					-	
	_		sg(%rbx), %xmm3	; xmm3 <- M[sq+rbx]	(0 0	-	1	-	-	-	-	-	-	-	1			-	-	
					0 0	0 00	-	-	-	-	-	-	-	-	-		-				
and the second s		moved	%xmm3, sq(%rbx)	; M[sq+rbx] <- xmm3	00	0 00	-	-	-	-	-	-	-	-		-					
				; ecx <- ecx - 1		1 1	-	-	-	-	-	-	-	-		- 3			-	_	
	_	000	\$8, %rax	; rax <- rax - 8	1	0 30	-	11		-	-	-	-			-	-		-		
	Etapas LOAD cauce Mul Se procesant 1 vez Mul blas instruc. 1 a 12 LD ADD/SVB A	Etapas LOAD 3 cauce MW 4 Se procesan 5 1 vez MW 6 Blas instruc. 7 1 a 12 LD 8 ADD/SVB 10 ADD/SVB 11 ADD/SVB 12 bits válido 1 ventana(s) 2 instrucciones 3 o 4 estación o 5 estaciones 6 de reserva (al finalizar 8 el ciclo) 9	Etapas (04) 3 moved cauce Mul 4 mulsd Se processan 5 moved 1 vez Mul 6 mulsd Blas instruc. 7 addsd 1 a 12 1 8 moved ADD/CVB 9 addsd ADD/CVB 11 dec ADD/CVB 12 subq bits válido 1 xorq ventana(s) 2 addq instrucciones 3 moved o 4 mulsd estación o 5 moved estaciones 6 mulsd de reserva 7 addsd (al finalizar 8 moved el ciclo) 9 addsd 10 moved 11 dec	Etapas 1940 3 moved x(\(\frac{1}{2}\) \(\frac{1}{2}\) \(Etapas 1940 3 moved x(\frax), \frac{\frax}{\frax} ; xmml <- M[x+rax] cauce Mul 4 mulsd \frac{\frax}{\frax} xmml ; xmml <- xmml <- xmml * xmml \frac{\frax}{\frac{xmml}{xmml}} * xmml <- xmml * xmml \frac{xmml}{xmml} * xmml	Etapas () 3 moved	Etapas Oab 3 movsd x(\frax), \frac{xmml}{xmml} ; xmml <- M[x+rax] F N	Etapas 1046 3 moved x(%rax), %xmml ; xmml <- M[x+rax] F h cauce hul 4 mulsd %xmml, %xmml ; xmml <- xmml * xmml F h lb Se procesan 5 moved y(%rax), %xmm0 ; xmm0 <- M[y+rax] LF h lb lo lo lo lo lo lo lo	Etapas 104) 3 moved	Etapas () 3 movsd	Etapas Oab 3 moved	Etapas (A) 3 moved	Etapas (A) 3 moved x(\{\frax\}, \{\frax\}, \{\frax\}, \{\frax\}\] ; xmml ; xmml \(-\text{Min} \) \(-\text{Min} \) \(\frac{1}{2} \) \(\frac{1}{	Etapas 10ab 3 movsd x(\{\frax\}, \{\frax\}\), \{\frac{xmml}{xmml}} \{\fr	Etapas Dab 3 moved x(\(\frac{tax}\), \(\frac{txmn}\) ; \(xmm\) < M(\(\frac{t+rax}\)) P D EX	Etapas 0ab 3 moved x(\{\text{rax}\), \{\text{xmml}\}; \{\	Etapas 0a) 3 moved x(%rax), %xmml; xmml; xmml <- M(x+rax) F D EX	Etapas Dah aucoe Mui 4 mulsd %xmml, %xmml; xmml <- M[x+rax] P D EX EX EX EX EX Se processing	Etapas Dah 3 movsd x(\frax), \frac{xmml}{xmml}; \frac{xmml}{xmml} < - \text{M(x+rax)} \text{IP} \t	Etapas OAb cauce Mul	Etapas OAA (*rax), *xmml ; xmml <- M[x+rax] P D EX

verano mpleo.



WUOLAH

	instrucción	operandos	explicación	1	2	3	4	5	6	7 8	9	10	11	12	12	14 1		1
	2 addg	%rbx, %rbx	/ rbx <- rbx xor rbx	IP	15	EX					1	-		12	13	14 1	2 16	A
Etapas	3 moved	\$N, %rax	/ rax <- rax + N	12	15													А
cauce	4 mulsd	x(%rax), %xmml	Xmml <= M[v4rav]	l ou	1.5		EK	EX									-	4
Se procesan	5 moved	%xmm1, %xmm1	; xmm1 <- xmm1 * xmm1	IF	13				25	EX 8	K E							
1 vez	6 mulad	y(%rax), %xmm0	1 xmm0 <- M[y+rax]		112	13			EX									
les instruc.	7 addsd	%xmmO, %xmmO	# xmm0 <- xmm0 * xmm0		111	15						EX	(bx	bk.	XX			
1 a 12	8 moved	%xmm1, %xmm0	1 xmm0 <- xmm0 + xmm1			13										EX		
	9 addsd	sq(%rbx), %xmm3	; xmm3 <- M[sq+rbx])lz	13	B			503	XE	X						
	10 moved	%xmm0, %xmm3	; xmm3 <- xmm3 + xmm0				15										EX	
		%xmm3, sq(%rbx)	: M[sq+rbx] <- xmm3			严	13											EX
	11 dec	%ecx	; ecx <- ecx - 1			III	di	EX										
	12 subq	\$8, %rax	; rax <- rax - 8			IF	ID	EX									1	
bits válido	1 xorg Abbyus	%rbx, %rbx	; rbx <- rbx xor rbx	13	31	35	-	11	34	31	1			,	-	-	-	-
ventana(s)	2 addg Abb/sut	\$N, %rax	; <u>rax</u> <- rax + N	_	31				31	31	-		-		-	-	-	-
instrucciones	3 moved Lb	x(%rax), %xmml	; _xmm1 <- M[x+rax]	1	0	_	_	1	4	4	-	-	-			-	-	-
0	4 mulsd MyL	%xmm1, %xmm1	; xmml <- xmml * xmml	00	00			00	11	11	-	-	-		-		-	-
estación o	5 moved L)	y(%rax), %xmm0	; xmm0 <- M[y+rax]	_	00	1	73	3	A	1	-		45				-	
estaciones	6 mulsd Mai	%xmm0, %xmm0	; xmm0 <- xmm0 * xmm0	0 0	00	200	73	00	00	00	21	53 .	-			-	-	1
de reserva	7 addsd Abb	%xmm1, %xmm0	; xmm0 <- xmm0 + xmm	_	0 00	_	_	00	00	00	00	00 0	LC	03 0	72 C	1 3:	-	
(al finalizar			xmm3 <- M[sq+rbx]		0 0	9	1-1	1	9	d	d	4.	世 -	4	4	d -	-	-
el ciclo)		%xmm0, %xmm3	; xmm3 <- xmm3 + xmm	0 0	0 09	00	10	00	00	00	00	00 1	20	如	20 :	5 11	0 33	-
			; M[sqtrbx] <- xmm3	0	0 00	30	20	30	10	30	10	30	ot	30	30	あり	0 30	3 3
		%ecx	; ecx <- ecx - 1		1 3	1	Tu	3	+	4	+	+	4	1 -	4 -	7 -		-
		\$8, %rax	; rax <- <u>rax</u> - 8	1	0 30	2 1	10	31	4	- 54	71	*	34	44	北	44 -		-
J UF PAI	A CARGA (ZC) RA STORE (JC) BD/SJD (ZC)		VENTANA CENTRALTZ				TILS MISIC			/c	CAMS	A						X
3 UF 1	AUL LYC)																	

WUOLAH

	instrucció	noperandos	explicación			_		- 1		- 1	- 1				2 40		1 1	
	1 xorq	%rbx, %rbx	; rbx <- rbx xor rbx	1				5	6	7	8	9	10 1	11 1	2 13	14	15	16
	2 addq	\$N, %rax	; rax <- rax + N	IF			DW											
Etapas	3 movsd	x(%rax), %xmm1	; xmm1 <- M[x+rax]			EX	WA	-1	uch									
cauce	4 mulsd	%xmm1, %xmm1	; xmm1 <- xmm1 * xmm1	12	10		EX	EX			her.	EX	414					
Se procesan	5 movsd	y(%rax), %xmm0	; xmm0 <- M[y+rax]	11	1F	15	1-4	EX	50	BK	υ×		NR					
1 vez	6 mulsd	%xmm0, %xmm0	; xmm0 <- xmm0 * xmm0		11		KA	EX	144	Ine	1-1		WK					
	7 addsd	%xmm1, %xmm0	; xmm0 <- xmm0 + xmm1		12	10			FR	EX	ES		EX	Wh				-
las instruc. 1 a 12 bits válido ventana(s) instrucciones	8 movsd	sq(%rbx), %xmm3	; xmm3 <- M[sq+rbx]		1p	10			EX	1=X				WA			100	
	9 addsd	%xmm0, %xmm3	; xmm3 <- xmm3 + xmm0		14	IE	IN			100				EX	aw			1
	10 movsd	%xmm3, sq(%rbx)	; M[sq+rbx] <- xmm3			11	11.3	1000							EX	X I	NB	1
	11 dec	%ecx	; ecx <- ecx - 1			IP		EX									VD	1
	12 subq	\$8, %rax	; rax <- rax - 8			11	110	EX	-							1	du	
bits válido	1 xorq AND	%rbx, %rbx	; rbx <- rbx xor rbx	1	1 32	-	7	-	-	-	-	-	1	,	-	-	-	-
ventana(s)	2 addg AND	\$N, %rax	; rax <- rax + N		1 30		-	-	-	-	-	-	-	-	-	-		-
Desired Control of the Control of th		x(%rax), %xmml	; xmm1 <- M[x+rax]	_	0 0		1	士	-	-	-	-	-	-	_	-	-	-
	4 mulsd Mun	%xmml, %xmml	; xmm1 <- xmm1 * xmm	1 0	0 00	00	00	00	31									
estación o	102	y(%rax), %xmm0	; xmm0 <- M[y+rax]		0 0			-	_	_			-	-	-	-		_
estaciones	6 mulsd Mul	%xmm0, %xmm0	; xmm0 <- xmm0 * xmm		0 00		000		111		-		-	-	_	-	-	-
de reserva	7 addsd Abb	%xmm1, %xmm0	; xmm0 <- xmm0 + xmm	1 0	0 0	000	0	00	00	00	00		31	-	-	-	-	-
(al finalizar	8 moved CARL	sq(%rbx), %xmm	3; xmm3 <- M[sq+rbx]		_	_	0 1	_				_		-	-	-	-	-
al ciclol	9 addsd AMD	%xmm0, %xmm3	; xmm3 <- xmm3 + xmm		00							2 20		4.7	-	-	-	
61 01010)	10 moved Cross	%xmm3, sq(%rbx)	; M[sqtrbx] <- xmm3	0	0 0	90	0 3			0 8	0 3	0 10	100	100	33	-		
	11 dec Abb	%ecx	; ecx <- ecx - 1		1 3	1 -	-	. 5	_	-			-	-	-	-	-	
	12 subq AbA	\$8, %rax	; rax <- ray - 8		10 9	0 3	D 3	1 5	1									200
ESPECIALCACION ES				ventv	ANA			iliza!	Sr.			North						
2 UF	CARGA (70)			CAIS	TAR,	DE	CONIFI	CAR	Y		EMIT	-ia	ч	15/2				
Z UF	ADDISOD (DC)																	
2 VF	MUL (46)																	