Código genera	ado	por gcc o	con -O2 para Intel x86																
		instruc.	operandos	explicación	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1	movsd	(%rbx,%rax,8), %xmm0	; xmm0<-M[rbx+rax*8]															
	2	movsd	8(%rbx,%rax,8), %xmm1	; xmm1<-M[8+rbx+rax*8]															
Etapas	3	movsd	16(%rbx,%rax,8), %xmm2	; xmm2<-M[16+rbx+rax*8]															
cauce	4	movsd	24(%rbx,%rax,8), %xmm3	; xmm3<-M[24+rbx+rax*8]															
Se procesan	5	addsd	%xmm0, %xmm4	; xmm4=xmm0+xmm4															
1 vez	6	addsd	%xmm1, %xmm5	; xmm5=xmm1+xmm5															
las instruc.	7	addsd	%xmm2, %xmm6	; xmm6=xmm2+xmm6															
1 a 11	8	addsd	%xmm3, %xmm7	; xmm7=xmm3+xmm7															
	9	addq	\$4, %rax	; rax=rax+4															
	10	cmpq	%rax, %rbp	; rbp-rax															
bits válido	1	movsd	(%rbx,%rax,8), %xmm0	; xmm0<-M[rbx+rax*8]	11														
ventana(s)	2	movsd	8(%rbx,%rax,8), %xmm1	; xmm1<-M[8+rbx+rax*8]	11														
instrucciones	3	movsd	16(%rbx,%rax,8), %xmm2	; xmm2<-M[16+rbx+rax*8]	11														
О	4	movsd	24(%rbx,%rax,8), %xmm3	; xmm3<-M[24+rbx+rax*8]	11														
estación o	5	addsd	%xmm0, %xmm4	; xmm4=xmm0+xmm4	01														
estaciones	6	addsd	%xmm1, %xmm5	; xmm5=xmm1+xmm5	01														
de reserva	7	addsd	%xmm2, %xmm6	; xmm6=xmm2+xmm6	01														
(al finalizar	8	addsd	%xmm3, %xmm7	; xmm7=xmm3+xmm7	01														
el ciclo)	9	addq	\$4, %rax	; rax=rax+4	11														
	10	cmpq	%rax, %rbp	; rbp-rax	01														
•		·																	
	_																		

addq	\$4, %rax	Bits de válido inicialmente a uno (11) porque el operando 4 está disponible y también está disponible (válido) el contenido de eax
movsd	16(%rbx,%rax,8), %xmm2	2 bit de válido porque usa dos registros con datos de entrada (rbx y rax) que se usan para obtener la dirección a la que se va a acceder para leer (la dirección se obtine con el cálculo 16+rbx+rax*8). En xmm2 se carga el dato que se trae de memoria. El primer bit se refiere a rbx y el segundo a rax (siguiendo el orden en el que aparecen en la instrucción)
addsd	%xmm0, %xmm4	Bits de válido inicialmente 01 porque el operando en %xmm4 está disponible pero el operando en %xmm0 no lo está (se obtiene como resultado de (1))

Posible pregunta en el control : ¿Qué instrucciones se están ejecutando en el ciclo 6? Para contestar a las preguntas hay que rellenar la tabla primero.