

## QUIZ 5

CONVOLUCION EN UNA DIMENSION E IMPLEMENTACION CON DIFERENTES  
TIPOS DE MEMORIA

ALUMNO: JOSE WILSON CASTAÑO MARIN

PROFESOR: JHON OSORIO RIOS

## INTRODUCCION

La convolución es una operación muy común en el área de procesamiento de señales. Por ejemplo puede ser utilizada para difuminar o realzar los bordes de una imagen, o suavizar los picos de amplitud en una señal de audio.

Que es la convolución y como la podemos aplicar en una dimensión?

Es operación que consiste en remplazar cada elemento de un array (1D) (este puede representar un pixel o un valor de volumen en la señal de audio) por una suma ponderada de sus vecinos. Los pesos con los que se pondera la suma están definidos por una máscara (un array de tantos elementos como vecinos a sumar).

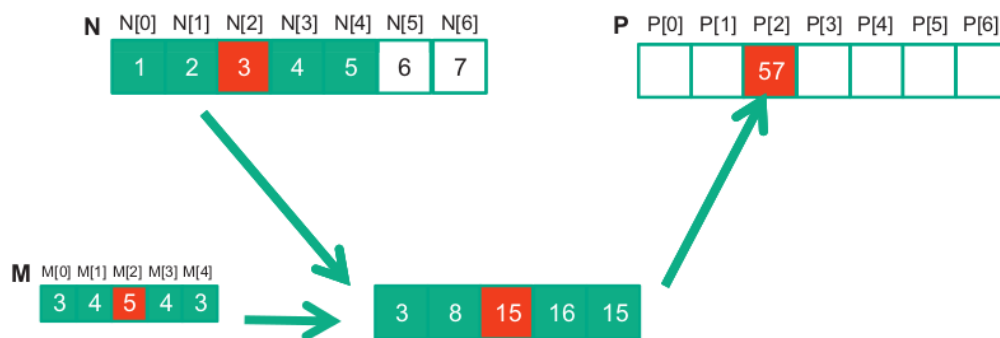


Figura 1: En la figura se muestra el cálculo del elemento

P[2]. El array de entrada es N y la máscara es M.

$$P[2] = M[0] \cdot N[0] + M[1] \cdot N[1] + M[2] \cdot N[2] + M[3] \cdot N[3] + M[4] \cdot N[4] = 57$$

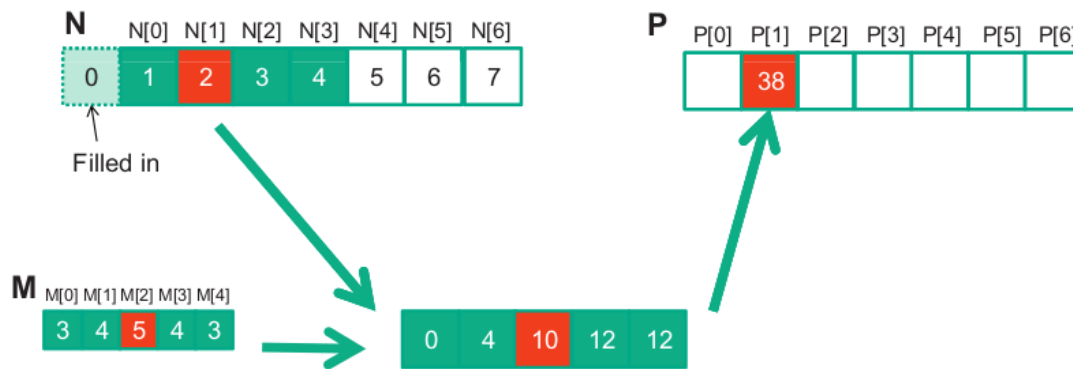


Figura 2: El cálculo de  $P[1] = 0 + 4 + 10 + 12 + 12 = 38$

## CONCLUSIONES

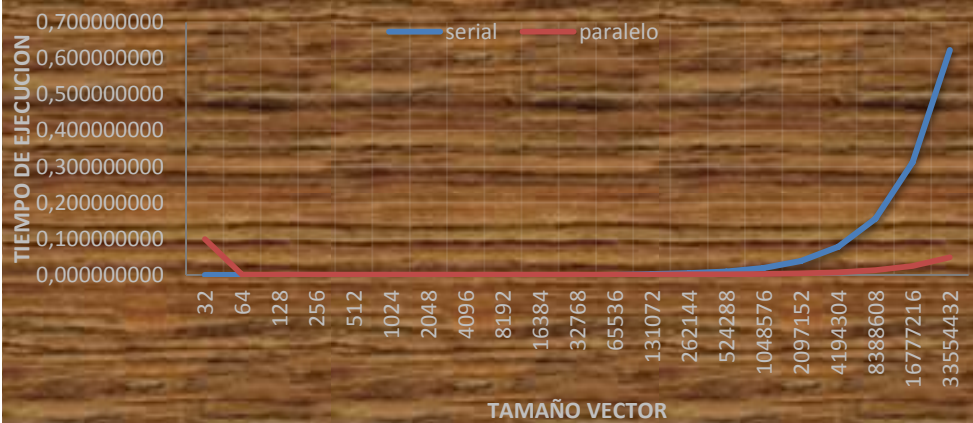
- La convolución es una operación atractiva para ser realizada por la GPU, debido a que cada suma requiere únicamente la lectura de los elementos actuales del vector o arreglo y la máscara, cada suma es independiente del resto y que cada escritura se realiza en posiciones distintas de la memoria.
- El manejo de bordes o elementos de las esquinas tales como de  $P[0]$ ,  $P[1]$ ,  $P[5]$  y  $P[6]$  involucran elementos que no forman parte del vector o arreglo, estos elementos tienen una importante incidencia en el desempeño del algoritmo y debemos tener cuidado con el tipo de memoria a utilizar con los elementos de relleno.
- Al realizar la convolucion al vector de entrada  $N$  con la Máscara  $M$  obtengo como resultado un nuevo vector  $P$ :

22	38	57	76	95	90	74
----	----	----	----	----	----	----

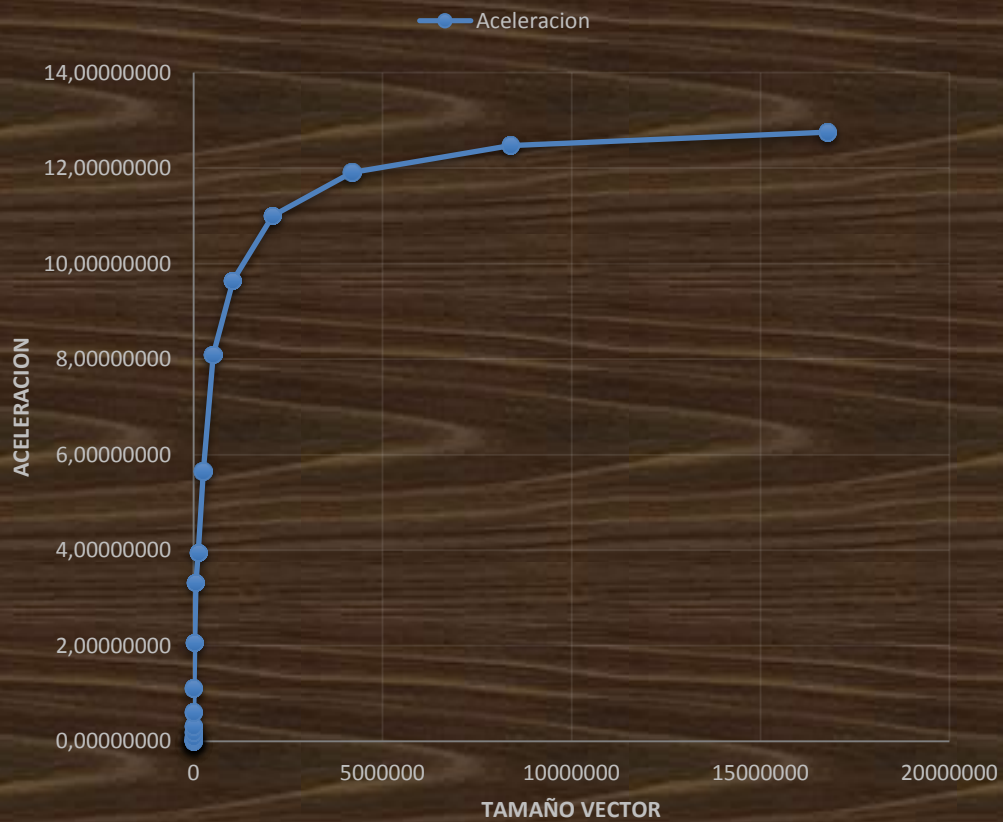
- Anexo archivo DATOSYGRAFICOS en Excel explicando resultados de los algoritmos con la convolucion en 1D.
- En la gráfica comparativa de las aceleraciones podemos observar que en algunos rangos de tiempo es mas optimo utilizar paralelizacion en memoria constante específicamente en los bordes.

DATOS CONVOLUCION CODIGO PARALELO BASICO					
Valor i	Ejecucion	Tamaño Vector	Tiempo Secuencial	Tiempo Paralelo	Aceleracion
5	0	32	0,000002000	0,099035	0,00002019
6	1	64	0,000001000	0,0002920	0,00342466
7	2	128	0,000003000	0,0002670	0,01123596
8	3	256	0,000005000	0,0002650	0,01886792
9	4	512	0,000009000	0,0004560	0,01973684
10	5	1024	0,000020000	0,0004800	0,04166667
11	6	2048	0,000038000	0,0004500	0,08444444
12	7	4096	0,000086000	0,0004600	0,18695652
13	8	8192	0,000152000	0,0004750	0,32000000
14	9	16384	0,000304000	0,0004960	0,61290323
15	10	32768	0,000606000	0,0005450	1,11192661
16	11	65536	0,001251000	0,0006060	2,06435644
17	12	131072	0,002447000	0,0007370	3,32021710
18	13	262144	0,004878000	0,0012350	3,94979757
19	14	524288	0,009732000	0,0017200	5,65813953
20	15	1048576	0,019451000	0,0024020	8,09783514
21	16	2097152	0,039141000	0,0040560	9,65014793
22	17	4194304	0,077802000	0,0070700	11,00452617
23	18	8388608	0,155764000	0,0130780	11,91038385
24	19	16777216	0,311693000	0,0249870	12,47420659
25	20	33554432	0,623559000	0,0488970	12,75250015

## Convolucion Secuencial y Paralelo Basico



## Aceleracion Paralelo Basico

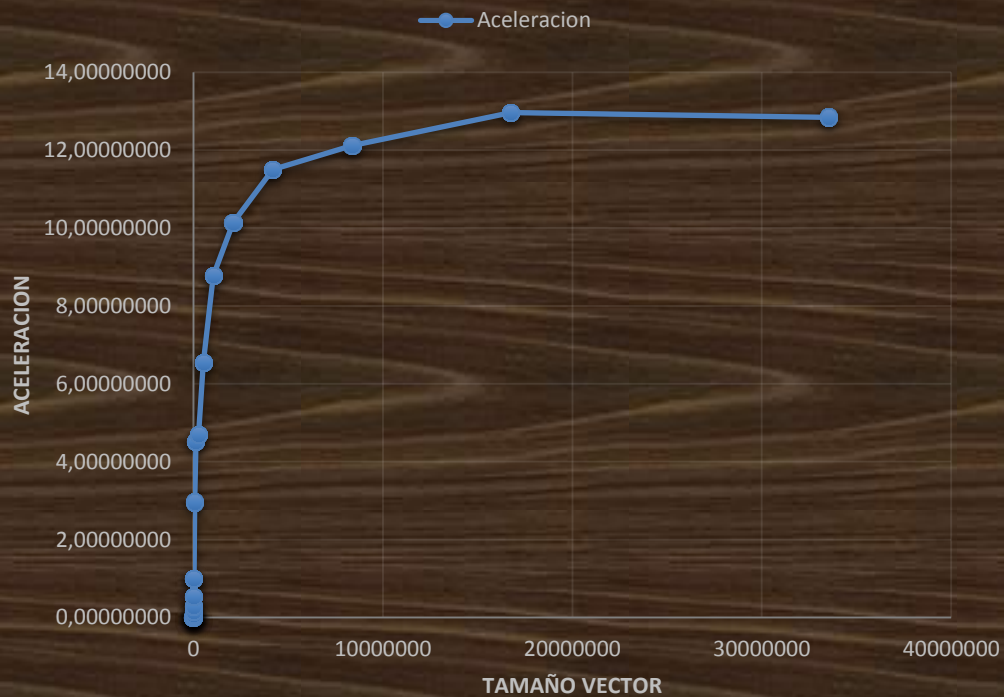


DATOS CONVOLUCION CODIGO PARALELO CON MEMORIA CONSTANTE					
Valor i	Ejecucion	Tamaño Vector	Tiempo Secuencial	Tiempo Paralelo	Aceleracion
5	0	32	0,000002000	0,095327	0,00002098
6	1	64	0,000002000	0,0002950	0,00677966
7	2	128	0,000003000	0,0002800	0,01071429
8	3	256	0,000005000	0,0002640	0,01893939
9	4	512	0,000009000	0,0002650	0,03396226
10	5	1024	0,000019000	0,0002690	0,07063197
11	6	2048	0,000038000	0,0002660	0,14285714
12	7	4096	0,000077000	0,0002730	0,28205128
13	8	8192	0,000151000	0,0002800	0,53928571
14	9	16384	0,000304000	0,0003040	1,00000000
15	10	32768	0,000628000	0,0006280	1,00000000
16	11	65536	0,001214000	0,0004110	2,95377129
17	12	131072	0,002449000	0,0005420	4,51845018
18	13	262144	0,004875000	0,0010410	4,68299712
19	14	524288	0,009739000	0,0014890	6,54063130
20	15	1048576	0,019458000	0,0022181	8,77237275
21	16	2097152	0,038865000	0,0038360	10,13164755
22	17	4194304	0,078819000	0,0068530	11,50138625
23	18	8388608	0,155822000	0,0128640	12,11302861
24	19	16777216	0,311023000	0,0240000	12,95929167
25	20	33554432	0,622520000	0,0484820	12,84022936

## Convolucion Secuencial y Paralelo con memoria constante



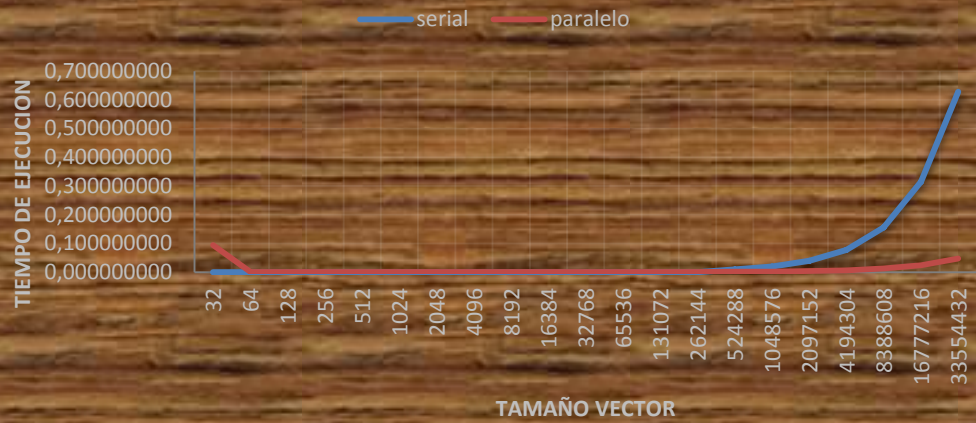
## Aceleracion Paralelo con Memoria Constante



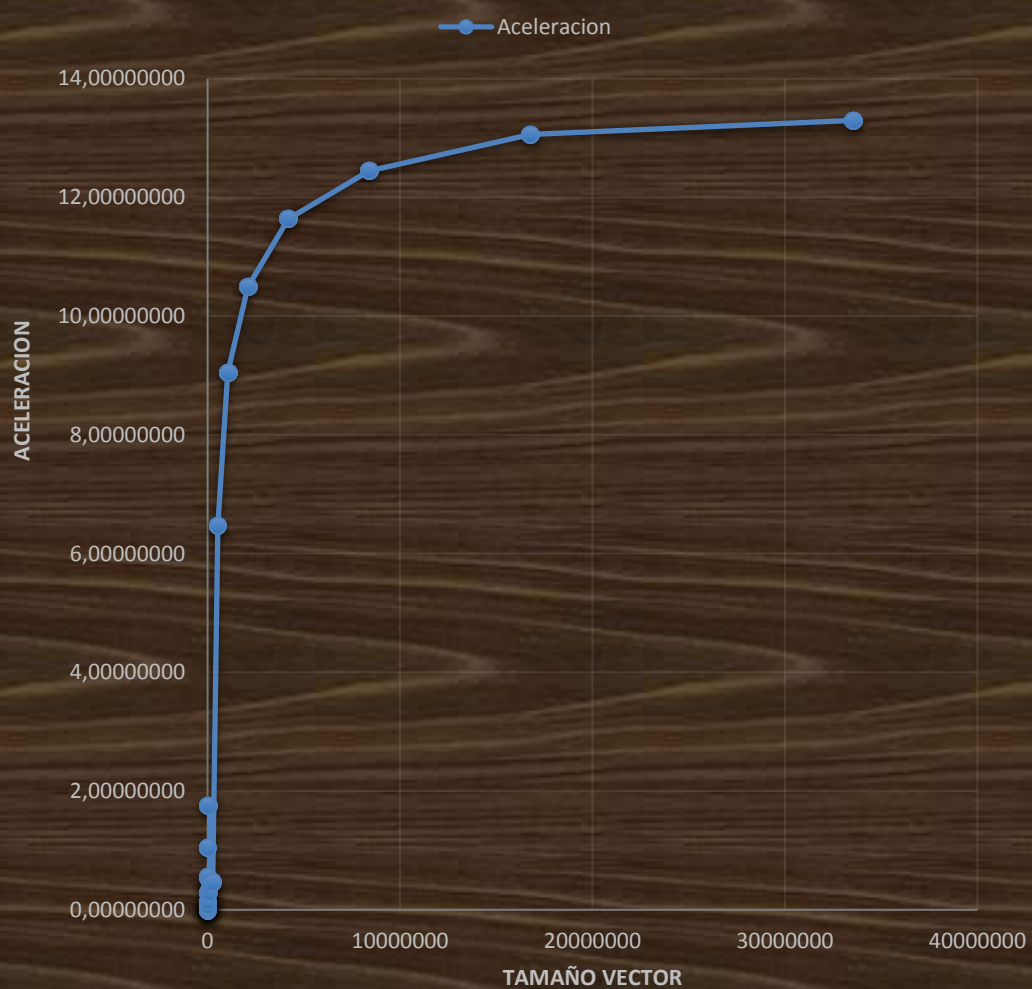
DATOS CONVOLUCION CODIGO PARALELO CON MEMORIA COMPARTIDA Y TILING					
Valor i	Ejecucion	Tamaño Vector	Tiempo Secuencial	Tiempo Paralelo	Aceleracion
5	0	32	0,000001000	0,09386	0,00001065
6	1	64	0,000002000	0,0002840	0,00704225
7	2	128	0,000003000	0,0002560	0,01171875
8	3	256	0,000005000	0,0002570	0,01945525
9	4	512	0,000010000	0,0002570	0,03891051
10	5	1024	0,000019000	0,0002650	0,07169811
11	6	2048	0,000038000	0,0002590	0,14671815
12	7	4096	0,000076000	0,0002630	0,28897338
13	8	8192	0,000152000	0,0002750	0,55272727
14	9	16384	0,000334000	0,0003160	1,05696203
15	10	32768	0,000608000	0,0003470	1,75216138
16	11	65536	0,000123700	0,0004110	0,30097324
17	12	131072	0,000244700	0,0005410	0,45231054
18	13	262144	0,000488400	0,0010480	0,46603053
19	14	524288	0,009752000	0,0015060	6,47543161
20	15	1048576	0,019518000	0,0021560	9,05287570
21	16	2097152	0,039626000	0,0037760	10,49417373
22	17	4194304	0,077955000	0,0066990	11,63681146
23	18	8388608	0,155954000	0,0125310	12,44545527
24	19	16777216	0,314717000	0,0240970	13,06042246
25	20	33554432	0,627789000	0,0472200	13,29498094



## Convolucion Secuencial y Paralelo con memoria constante



## Aceleracion Paralelo con Memoria Compartida y Tiling



100

