



Área Académica de Ingeniería en Computadores

Curso

CE 4302 – Arquitectura de Computadores II

Profesor

Ronald García Fernandez

Reference Sheet uARCH

Estudiantes:

2018225504 - Josue cubero

2019173506 - Brandon Gomez

2019031688- Leonardo Guillén

2018080231- Juan Peña

2020096499- Ignacio morales

II Semestre 2023

TABLA DE CONTENIDOS

Lista de instrucciones: quick reference	3
Codificación de instrucciones	4
Especificación de registros	7
Descripción de instrucciones	9
Segmentación de las memorias de la microarquitectura	11

Lista de instrucciones: quick reference

A continuación, las tablas 1 y 2 resumen el conjunto de instrucciones soportadas por la microarquitectura diseñada. La tabla 1 presenta un listado de instrucciones básicas de procesamiento de datos, memoria y saltos.

Este ISA está basado en RISC de 24 bits de longitud por instrucción adaptado para la generación de gráficos a color RGB, por lo que también utiliza datos en memoria y operandos de 24 bits.

Tabla 1. Resumen de instrucciones básicas del ISA.

Instrucción	Sintaxis	RTL	Mode Address
sum	sum Rd, Rn, Rm	$Rd \leftarrow Rn + Rm$	Register
res	res Rd, Rn, Rm	$Rd \leftarrow Rn - Rm$	Register
mul	mul Rd, Rn, Rm	$Rd \leftarrow Rn * Rm$	Register
div	div Rd, Rn, Rm	$Rd \leftarrow Rn / Rm$	Register
mod	sum Rd, Rn, Rm	$Rd \leftarrow Rn \% Rm$	Register
cmp	cmp Rn, Rm	$Rn - Rm$; FLAGS \leftarrow CVNZ	Register
mov	mov Rd, Rm	$Rd \leftarrow Rm$	Register
movi	movi Rd, Inm	$Rd \leftarrow Inm$	Inm
crg	crg Rd, Rn	$Rd \leftarrow MEM[Rn]$	Register
esc	esc Rm, Rn	$MEM[Rm] \leftarrow Rn$	Register
sig	sig label	PC \leftarrow branchAddress si comparación resulta en igual que.	conditional
slm	slm label	PC \leftarrow branchAddress si comparación resulta en menor que.	conditional
sali	sali label	PC \leftarrow branchAddress	Incondicional

La tabla 2 lista las instrucciones soportadas, ofrece una primera intuición sobre

Tabla 2. Resumen de instrucciones para la extensión vectorial.

Instrucción	Sintaxis	RTL	Modo Address
vsum	vsum VRd, VRn, VRm	VRd <- VRn + VRm	Register
vres	vres VRd, VRn, VRm	VRd <- VRn - VRm	Register
vand	vand VRd, VRn, VRm	VRd <- VRn * VRm	Register
vcmlt	vcmlt VRd, VRn, VRm	VRd <- VRn < VRm	Register
vmov	vmov VRd[#], Rm	VRd[#] <- Rm	Register
vmovf	vmovf VRd, VRm	VRd[#] <- Rm	Register
vcrg	vcrg VRd, VRn	VRd[0] <- MEM[Rn] VRd[1] <- MEM[Rn+1] VRd[2] <- MEM[Rn+2] ..etc	Registro
vesc	vesc Rm, VRn	MEM[Rm] <- Rn MEM[Rm] <- VRn[0] MEM[Rm+1] <- VRn[1] MEM[Rm+2] <- VRn[2] ... etc	Registro

Codificación de instrucciones

Tabla 3. INSTRUCCIONES DE DATOS ESCALARES

Mnemonic	OPCODE [23:20]	RD[19:16]	RN[15:12]	RM[11:8]	FREE[7:5]	isVector[4]	vect_esc[3]	FREE[2:0]
sum	0101	RD	RN	RM		0	0	
res	0110	RD	RN	RM		0	0	
mul	1000	RD	RN	RM		0	0	
div	1001	RD	RN	RM		0	0	
mod	1110	RD	RN	RM		0	0	
cmp	1010		RN	RM		0	0	

mov	0011	RD	RN			0	0	
movi	0010	RD	inm					
out	0100		RN			0	0	

En la tabla 3 se observan los distintos componentes de la decodificación de las instrucciones de datos escalares, a continuación se muestra una descripción de cada una:

1. Mnemónico: Nombre clave de la instrucción
2. OPCODE: identificador principal de la instrucción
3. RD: Registro Destino
4. RN: Operando 1
5. RM: Operando 2
6. FREE: Bits que no se utilizan
7. isVector: bit que indica si la instrucción es vectorial o escalar.
8. Vec_esc: bit que indica si la instrucción es Vectorial-Escalar

En el caso de la instrucción movi, se observa RN es un inmediato de tipo entero de 16 bits, por lo que toma los bits [15:0].

Tabla 4. INSTRUCCIONES DE DATOS VECTORIALES								
Mnemonico	OPCODE [23:20]	RD[19:16]	RN[15:12]	RM[11:8]	FREE[7:5]	isVector[4]	vect_esc[3]	INDEX[2:0]
vsum	0101	RD	RN	RM		1	0	
vres	0110	RD	RN	RM		1	0	
vmul	1000	RD	RN	RM		1	0	
vand	1001	RD	RN	RM		1	0	
vcmlpt	1110	RD	RN	RM		1	0	
vmov	0011	RD	RN			1	1	#
vmovf	0011	RD	RN			1	0	

En la tabla 4 se observa una columna llamada INDEX, esta sirve para indexar los vectores cuando se utiliza la operación “vmov”.

Tabla 5. INSTRUCCIONES DE MEMORIA

Mnemonico	OPCODE [23:20]	RD[19:16]	RN[15:12]	RM[11:8]	FREE[7:5]	isVector[4]	vect_esc[3]	FREE[2:0]
crg	0111	RD	RN			0	0	
esc	0001		RN	RM		0	0	

Tabla 6. INSTRUCCIONES DE MEMORIA VECTORIAL

Mnemonico	OPCODE [23:20]	RD[19:16]	RN[15:12]	RM[11:8]	FREE[7:5]	isVector[4]	vect_esc[3]	INDEX[2:0]
vcrg	0111	RD	RN			1	0	
vesc	0001		RN	RM		1	1	

Tabla 7. INSTRUCCIONES DE CONTROL

Mnemonico	OPCODE [23:20]	RD[19:16]	ETIQUETA[15:0]					
sig	1011		Imm					
sim	1101							
sali	1111							
sr	1100	RD						

En la tabla 7 se puede notar como se utiliza la columna ETIQUETA, este toma 16 bits y consiste en un inmediato utilizado para realizar saltos dentro del código, alterando el PC de manera relativa.

Especificación de registros

La tabla 8 ejemplifica la composición del Scalar Register File, donde cada registro almacena datos de 24 bits.

Tabla 8. Codificación del banco de registros escalares.

Banco de registros escalares		
ID	code	Proposito
R0	0000	Registro de Proposito General
R1	0001	
R2	0010	
R3	0011	
R4	0100	
R5	0101	
R6	0110	
R7	0111	
R8	1000	
R9	1001	
R10	1010	
R11	1011	
R12	1100	
R13	1101	
R14	1110	
R15	1111	PC

La tabla 9 ejemplifica la composición del Vector Register File, donde cada registro almacena datos de 192 bits..

Tabla 9. Codificación del banco de registros vectoriales.

Banco de registros vectoriales		
ID	code	Proposito
VR0	0000	Registro de Proposito General
VR1	0001	
VR2	0010	
VR3	0011	
VR4	0100	
VR5	0101	
VR6	0110	
VR7	0111	
VR8	1000	
VR9	1001	
VR10	1010	
VR11	1011	
VR12	1100	
VR13	1101	
VR14	1110	
VR15	1111	

Tabla 10. Contenido de un registro en particular para un vector length de 8.

VRd							
VRd[7]	VRd[6]	VRd[5]	VRd[4]	VRd[3]	VRd[2]	VRd[1]	VRd[0]
scalar7	scalar6	scalar5	scalar4	scalar3	scalar2	scalar1	scalar0

Descripción de instrucciones

Caracterización de instrucciones de datos

Tabla 11. Descripción de instrucciones de datos para operandos escalares.

Instrucción	Ejemplo	Descripción
sum	sum R2, R1, R0	La suma de los contenidos en los R1 y R0 se almacena en R2
res	res R5, R4, R3	La resta de los contenidos en los R4 y R3 se almacena en R5
mul	mul R8, R7, R6	La multiplicación de los contenidos en los R7 y R6 se almacena en R8
div	div R2, R2, R0	La división de los contenidos en R2 y R0 se almacena en R2
mod	mod R8, R7, R6	El módulo o residuo de una división de R7 con R6 se almacena en R6
cmp	cmp R1, R2	Realiza la diferencia entre los registros R1 y R2 y actualiza las banderas del procesador
mov	mov R5, R4	Realiza un bypass del valor guardado en R4 a R5
movi	movi R4, 233	Asigna el valor entero 233 a R4

Tabla 12. Descripción de instrucciones de datos para operandos vectoriales.

Instrucción	Ejemplo	Descripción
vsum	vsum VR3, VR2, VR1	Suma de los 8 valores que hay en cada registro vectorial VR1 y VR2
vres	vres VR5, VR4, VR3	Resta de los 8 valores que hay en cada registro vectorial VR1 y VR2
vand	vand VR6, VR11, VR2	Operación lógica AND de los 8 valores que hay en cada registro vectorial VR1 y VR2. Se utiliza para conservar el los datos/píxeles que deban de escribirse en la memoria, el resto de valores lo hace cero.
vcmplt	vcmplt VR9, VR5, VR4	Almacena en VR9 el resultado de evaluar que cada contenido en el vector VR5 sea menor que su contraparte en VR4
vmov	vmov VR9[4], R0	Asigna el valor escalar guardado en R0 a una posición en particular del vector VR9, en este caso 4.
vmovf	vmovf VR5, VR7	Asigna el valor vectorial guardado en VR7 a VR5

Caracterización de instrucciones de memoria

Tabla 13. Descripción de instrucciones memoria para operandos escalares.

Instrucción	Ejemplo	Descripción
crg	crg Rd, Rn	Se carga el contenido al registro Rd de lo que se encuentra en la dirección de memoria en Rn
esc	esc Rm, Rn	Se escribe en la dirección de memoria Rm el contenido de registro Rn.

Tabla 14. Descripción de instrucciones de memoria para operandos vectoriales.

Instrucción	Sintaxis	Descripción
vcrg	vcrg VR1, VR2	Se lee el contenido de la memoria de acuerdo con las 8 direcciones que hay en VR2 y sus datos son cargados al vector VR1
vesc	vesc R1, VR2	Se escribe el contenido del registro VR2 de acuerdo con las 8 direcciones continuas a partir de la dirección contenida en R1

Caracterización de instrucciones de control-salto

Tabla 15. Descripción de instrucciones datos para operandos vectoriales.

Instrucción	Ejemplo	Descripción
sig	sig label	Realiza un salto a la dirección donde esta el label si la comparación es mayor. Se realiza el salto en función de los valores en los flags.
sim	sim label	Realiza un salto a la dirección donde esta el label si la comparación es menor. Se realiza el salto en función de los valores en los flags.
sali	sali label	Realiza un salto incondicional a la dirección donde esta el label.

Acá, cada label se compone de un valor inmediato de 16 bits con complemento a 2. Luego se hace un corrimiento a la izquierda para realizar la extensión de la etiqueta a 24 bits y así sumarlo al PC actual.

Segmentación de las memorias de la microarquitectura

Se tienen 3 memorias:

- Memoria de colores: En el caso de la memoria de colores, estos se encuentran desde la direccion 0 a la 23.
- Memoria de pixeles: En el caso de la memoria de pixeles, se tienen las direcciones desde 24 a 10023.
- Memoria de instrucciones: En el caso de la memoria de instrucciones de instrucciones se tienen las direcciones desde 0 hasta 255.