

Nivel ISA – OMAP4430

Arquitectura y Organización de Computadoras II

Ticiano J. Torres Peralta

REV2021

OMAP4430

El OMAP4430 es un SoC (System on Chip) donde el procesador, el ARM Cortex A9, implementa el ISA ARM v7. Este mismo ISA existe desde los años 1980, a diferencia del i7, es una arquitectura RISC, que significa que es una arquitectura LOAD/STORE.

Tiene dos modos de operación:

- Kernel Mode
- User Mode

Modelo de Memoria

El modelo de memoria del ARM v7 es simple, un solo espacio de 32 bits (direcciones 0 a $2^{32} - 1$). A pesar de su simpleza, v7 trajo el problema que no puede diseccionar mas de 4 GB. Esto llevo a crear problemas en la industria de smartphones, porque rápidamente sus necesidades incrementaron mas allá que 4 GB. Programadores de sistema, encontraron una solución temporaria usando la memoria FLASH, como la gran parte de la memoria principal.

Es bi-endianness, que quiere decir que puede funcionar como big-endian o little-endian. Esto se configura a través de unos bloques de memoria de sistema, y el procesador se configura en el momento que inicia. Esta memoria de sistema siempre está en little-endian a pesar de como sea la configuración.

La memoria es alineada dado que las instrucciones son de longitud fija.

Formato de Instrucción

A mucha diferencia del formato de instrucción del IA-32, el ARM v7 es limpio. Esta tienen instrucciones de longitud fija, de 16 y 32 bits.

Las instrucciones de 16 bits son versiones cortas de las de 32 bits, permitiendo código un poco mas comprimido pero con limitaciones, por ejemplo, no puede diseccionar todos los registros. Este “sub ISA” es conocido como el Thumb ISA.

Formato de Instrucción

La dificultad en entender el formato de instrucción está en que los diseñadores hicieron todo lo posibles para utilizar cada bit disponible. Por esta razón, cada tipo de instrucción tiene un formato específico.

31	28	27					16	15					8	7					0	Instruction type				
Cond	0	0	I	Opcode			S	Rn	Rd	Operand2								Data processing / PSR Transfer						
Cond	0	0	0	0	0	0	A	S	Rd	Rn	RS	1	0	0	1	Rm	Multiply							
Cond	0	0	0	0	1	U	A	S	RdHi	RdLo	RS	1	0	0	1	Rm	Long Multiply							
Cond	0	0	0	1	0	B	0	0	Rn	Rd	0	0	0	0	1	0	0	1	Rm	Swap				
Cond	0	1	I	P	U	B	W	L	Rn	Rd	Offset				Load/Store Byte/Word									
Cond	1	0	0	P	U	S	W	L	Rn	Register List				Load/Store Multiple										
Cond	0	0	0	P	U	1	W	L	Rn	Rd	Offset1	1	S	H	1	Offset2	Halfword transfer: Immediate offset							
Cond	0	0	0	P	U	0	W	L	Rn	Rd	0	0	0	0	1	S	H	1	Rm	Halfword transfer: Register offset				
Cond	1	0	1	L	Offset								Branch											
Cond	0	0	0	1	0	0	1	0	1	1	1	1	1	1	1	0	0	0	1	Rn	Branch Exchange			
Cond	1	1	0	P	U	N	W	L	Rn	CRd	CPNum	Offset				Coprocessor data transfer								
Cond	1	1	1	0	Op1			CRn	CRd	CPNum	Op2	0	CRm			Coprocessor data operation								
Cond	1	1	1	0	Op1	L	CRn	Rd	CPNum	Op2	1	CRm			Coprocessor register transfer									
Cond	1	1	1	1	SWI Number								Software interrupt											

Formato de Instrucción

Vale la pena mencionar el campo COND, presente en todas las instrucciones. Este se llama el campo de condición y define si la instrucción se va a completar o no, dependiendo de las condiciones del sistema.

Puede ser útil para optimizar el rendimiento porque, normalmente en un salto condicional, uno no sabe cual es la próxima instrucción. Si la maquina tienen que esperar hasta que se complete la instrucción condicional, es un golpe grande al rendimiento. Con el campo COND, la próxima instrucción puede empezar a ejecutarse inmediatamente y decidir completar (escribir resultados a registros) según la condición de la previa.

Se usan 5 bits para el campo que direccionar registros, y 12 bits para el campo de valor inmediato.

Modos de Direccionamiento

Los modos soportados son:

- Register
- Immediate
- Displacement
- Register Indirect
- PC Relative

Los últimos tres modos de direccionamiento son exclusivamente para las instrucciones de tipo LOAD/STORE y saltos.

Tipos de Datos

- El ARM v7 del OMAP4430 soporta:
 - Enteros signados (codificado en complemento-a-dos).
 - Enteros no-signados
 - Punto flotante (codificado en IEEE 754)

Type	8 Bits	16 Bits	32 Bits	64 Bits
Signed integer	x	x	x	
Unsigned integer	x	x	x	
Binary coded decimal integer				
Floating point			x	x

El OMAP4430 tienen una manera particular para tratar números. Las instrucciones de lecturas/escrituras, indican tanto el tamaño y el signo del valor y después internamente extienden ese numero a un tamaño de 32 bits. De esta manera simplifica el camino de datos, ya que internamente solo trata con objetos de 32 bits.

No tienen soporte de hardware para caracteres o strings. Todo se maneja a nivel software.

Tipos de Instrucciones

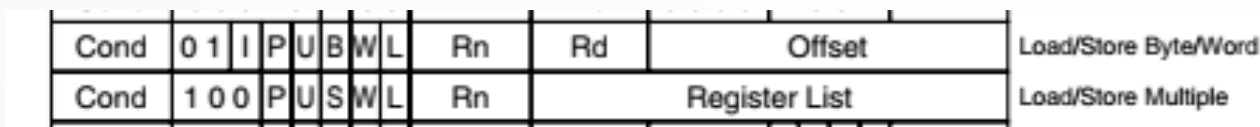
En términos de tipos de instrucciones, el ARM v7 tiene un set muy reducido en comparación al IA-32.

Las siguientes son casi todas las instrucciones disponibles en User Mode.

Tipos de Instrucciones

Instrucciones de LOAD/STORE vienen en solo tres sabores, para leer/escribir 1, 2, o 4 Bytes.

Formato de Instrucción



Loads

LDRSB DST,ADDR	Load signed byte (8 bits)
LDRB DST,ADDR	Load unsigned byte (8 bits)
LDRSH DST,ADDR	Load signed halfwords (16 bits)
LDRH DST,ADDR	Load unsigned halfwords (16 bits)
LDR DST,ADDR	Load word (32 bits)
LDM S1,REGLIST	Load multiple words

Stores

STRB DST,ADDR	Store byte (8 bits)
STRH DST,ADDR	Store halfword (16 bits)
STR DST,ADDR	Store word (32 bits)
STM SRC,REGLIST	Store multiple words

S1 = source register
S2IMM = source register or immediate
S3 = source register (when 3 are used)
DST = destination register
D1 = destination register (1 of 2)
D2 = destination register (2 of 2)

ADDR = memory address
IMM = immediate value
REGLIST = list of registers
PSR = processor status register
cc = branch condition

Tipos de Instrucciones

Instrucciones aritméticas tienen la posibilidad de modificar los códigos de condición del procesador, y por esta razón viene en dos formas. Una en que si lo modifican, y otra en que no. Esto permite al compilar poder reordenar de forma mas optima las instrucciones.

Formato de Instrucción

Cond	0 0	I	Opcode	S	Rn	Rd	Operand2			Data processing / PSR Transfer
Cond	0 0 0 0 0 0		A S		Rd	Rn	RS	1 0 0 1	Rm	Multiply
Cond	0 0 0 0 1	U	A S		RdHi	RdLo	RS	1 0 0 1	Rm	Long Multiply

S1 = source register
S2IMM = source register or immediate
S3 = source register (when 3 are used)
DST = destination register
D1 = destination register (1 of 2)
D2 = destination register (2 of 2)

ADDR = memory address
IMM = immediate value
REGLIST = list of registers
PSR = processor status register
cc = branch condition

Arithmetic

ADD DST,S1,S2IMM	Add
ADD DST,S1,S2IMM	Add with carry
SUB DST,S1,S2IMM	Subtract
SUB DST,S1,S2IMM	Subtract with carry
RSB DST,S1,S2IMM	Reverse subtract
RSC DST,S1,S2IMM	Reverse subtract with carry
MUL DST,S1,S2	Multiply
MLA DST,S1,S2,S3	Multiple and accumulate
UMULL D1,D2,S1,S2	Unsigned long multiple
SMULL D1,D2,S1,S2	Signed long multiple
UMLAL D1,D2,S1,S2	Unsigned long MLA
SMLAL D1,D2,S1,S2	Signed long MLA
CMP S1,S2IMM	Compare and set PSR

Tipos de Instrucciones

Instrucciones booleanas son análogas a las aritméticas

Formato de Instrucción



Boolean

TST DST,S1,S2IMM	Test bits
TEQ DST,S1,S2IMM	Test equivalence
AND DST,S1,S2IMM	Boolean AND
EOR DST,S1,S2IMM	Boolean Exclusive-OR
ORR DST,S1,S2IMM	Boolean OR
BIC DST,S1,S2IMM	Bit clear

S1 = source register
S2IMM = source register or immediate
S3 = source register (when 3 are used)
DST = destination register
D1 = destination register (1 of 2)
D2 = destination register (2 of 2)

ADDR = memory address
IMM = immediate value
REGLIST = list of registers
PSR = processor status register
cc = branch condition

Tipos de Instrucciones

Las instrucciones de desplazamiento son las mínimas, un shift a la izquierda (lógico) y dos a la derecha (lógico y aritmético). Una sola instrucción de desplazamiento circular.

Formato de Instrucción



Shifts/rotates

LSL DST,S1,S2IMM	Logical shift left
LSR DST,S1,S2IMM	Logical shift right
ASR DST,S1,S2IMM	Arithmetic shift right
ROR DSR,S1,S2IMM	Rotate right

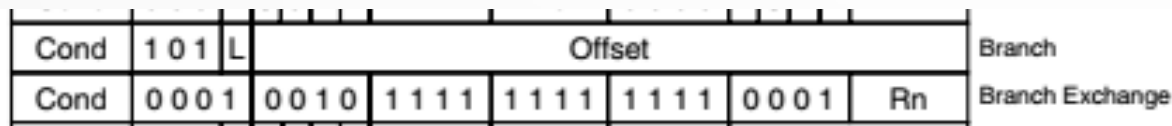
S1 = source register
S2IMM = source register or immediate
S3 = source register (when 3 are used)
DST = destination register
D1 = destination register (1 of 2)
D2 = destination register (2 of 2)

ADDR = memory address
IMM = immediate value
REGLIST = list of registers
PSR = processor status register
cc = branch condition

Tipos de Instrucciones

Instrucciones de control de flujo incluyen una instrucción que salto condicional que maneja todas las condiciones, y dos para llamada de procedimientos.

Formato de Instrucción



Transfer of control

Bcc IMM	Branch to PC+IMM
BLcc IMM	Branch with link to PC+IMM
BLcc S1	Branch with link to reg add

S1 = source register
S2IMM = source register or immediate
S3 = source register (when 3 are used)
DST = destination register
D1 = destination register (1 of 2)
D2 = destination register (2 of 2)

ADDR = memory address
IMM = immediate value
REGLIST = list of registers
PSR = processor status register
cc = branch condition

Tipos de Instrucciones

Miscellaneous

MOV DST,S1	Move register
MOVT DST,IMM	Move imm to upper bits
MVN DST,S1	NOT register
MRS DST,PSR	Read PSR
MSR PSR,S1	Write PSR
SWP DST,S1,ADDR	Swap reg/mem word
SWPB DST,S1,ADDR	Swap reg/mem byte
SWI IMM	Software interrupt

S1 = source register

S2IMM = source register or immediate

S3 = source register (when 3 are used)

DST = destination register

D1 = destination register (1 of 2)

D2 = destination register (2 of 2)

ADDR = memory address

IMM = immediate value

REGLIST = list of registers

PSR = processor status register

cc = branch condition