Projet SI3 - PEP

Synthèse des instructions ARM à implémenter

Parmi toutes les instructions du jeu ARM v7, vous devrez implémenter les 4 types d'instructions suivants:

a) Shift, add, sub, mov,
   - 6 instructions

b) Data Processing,
   - 16 instructions

c) Load/Store,
   - 2 instructions

d) Branch
   - 1 instruction

Chaque instruction est codée sur 16 bits dans un format propre à chaque type. Pour différencier lez types d'instructions, le jeu d'instructions ARM prévoit un code d'instruction spécifique dans les bits de poids forts :

Code d'instruction		Catégorie A	Catégorie B	Catégorie C	Catégorie D
00 XX XX	Shift, add, sub…	1			
01 00 00	Data processing		1		
01 10 XX	Load/Store			1	
11 01 XX	Branch				1

Chaqye catégorie dispose de son propre format de codage que vous retrouverez dans la documentation ARM (ARMv7-M Architecture Reference Manual 2014, chapitre A5, disponible sur jalon).

Quand plusieurs codages sont proposés pour une instruction, vous choisirez le codage T1 (sur 16 bits). On notera cependant deux exceptions pour les instructions LDR et STR ou vous choisirez entre le codage T1 ou le codage T2 en justifiant votre choix.

Table A5-1 16-bit Thumb instruction encoding

opcode	Instruction or instruction class
00xxxx	*Shift (immediate), add, subtract, move, and compare* on page A5-128
010000	*Data processing* on page A5-129
010001	*Special data instructions and branch and exchange* on page A5-130
01001x	Load from Literal Pool, see *LDR (literal)* on page A7-254
0101xx 011xxx 100xxx	*Load/store single data item* on page A5-131
10100x	Generate PC-relative address, see *ADR* on page A7-197
10101x	Generate SP-relative address, see *ADD (SP plus immediate)* on page A7-193
1011xx	*Miscellaneous 16-bit instructions* on page A5-132
11000x	Store multiple registers, see *STM, STMIA, STMEA* on page A7-422
11001x	Load multiple registers, see *LDM, LDMIA, LDMFD* on page A7-248
1101xx	*Conditional branch, and supervisor call* on page A5-134
11100x	Unconditional Branch, see *B* on page A7-207

Table A5-2 16-bit shift (immediate), add, subtract, move and compare encoding

opcode	Instruction	See
000xx	Logical Shift Left[a]	*LSL (immediate)* on page A7-298
001xx	Logical Shift Right	*LSR (immediate)* on page A7-302
010xx	Arithmetic Shift Right	*ASR (immediate)* on page A7-203
01100	Add register	*ADD (register)* on page A7-191
01101	Subtract register	*SUB (register)* on page A7-450
01110	Add 3-bit immediate	*ADD (immediate)* on page A7-189
01111	Subtract 3-bit immediate	*SUB (immediate)* on page A7-448
100xx	Move	*MOV (immediate)* on page A7-312
101xx	Compare	*CMP (immediate)* on page A7-229
110xx	Add 8-bit immediate	*ADD (immediate)* on page A7-189
111xx	Subtract 8-bit immediate	*SUB (immediate)* on page A7-448

Table A5-3 16-bit data processing instructions

opcode	Instruction	See
0000	Bitwise AND	*AND (register)* on page A7-201
0001	Exclusive OR	*EOR (register)* on page A7-239
0010	Logical Shift Left	*LSL (register)* on page A7-300
0011	Logical Shift Right	*LSR (register)* on page A7-304
0100	Arithmetic Shift Right	*ASR (register)* on page A7-205
0101	Add with Carry	*ADC (register)* on page A7-187
0110	Subtract with Carry	*SBC (register)* on page A7-380
0111	Rotate Right	*ROR (register)* on page A7-368
1000	Set flags on bitwise AND	*TST (register)* on page A7-466
1001	Reverse Subtract from 0	*RSB (immediate)* on page A7-372
1010	Compare Registers	*CMP (register)* on page A7-231
1011	Compare Negative	*CMN (register)* on page A7-227
1100	Logical OR	*ORR (register)* on page A7-336
1101	Multiply Two Registers	*MUL* on page A7-324
1110	Bit Clear	*BIC (register)* on page A7-213
1111	Bitwise NOT	*MVN (register)* on page A7-328

Table A5-5 16-bit Load/store instructions

opA	opB	Instruction	See
0101	000		
0101	001		
0101	010		
0101	011		
0101	100		
0101	101		
0101	110		
0101	111		
0110	0xx	Store Register	*STR (immediate)* on page A7-426
0110	1xx	Load Register	*LDR (immediate)* on page A7-252
0111	0xx		
0111	1xx		
1000	0xx		
1000	1xx		
1001	0xx		
1001	1xx		

Table A5-8 Branch and supervisor call instructions

opcode	Instruction	See
not 111x	Conditional branch	*B* on page A7-207
1110		
1111		