YARP Instruction set architecture

2016-11-28 15:51

Byte#	0	1	2	3	4	5
halt	0	0				
nop	1	0				
mov rA,rB	2	0	rA	rB		
mov rA,#imm32	3	0	rA	F	imm32	
ldr rA,[rB]	4	0	rA	rB		
str rA,[rB]	4	1	rA	rB		
ldr rA,[rB,#imm32]	5	0	rA	rB	imm32	
str rA,[rB,#imm32]	5	1	rA	rB	imm32	
OP rA,rB	6	fn	rA	rB		
b*cond* adrs	7	fn	absolute adrs			
call adrs	8	0	absolute adrs			
ret	9	0				
push rA	A	0	rA	F		
pop rA	B	0	rA	F		

Reg. ID	Name
0	r0
1	r1
2	r2
3	r3
4	r4
5	r5
6	r6
7	r7 or sp
F	none

Prog ctr:	PC

Operation *OP* depends on the 4-bit OP fn code.

fn	OP
0	add
1	sub
2	and
3	eor
...	...

All operations set the flags

Branch condition depends on the 4-bit branch fn-code

fn	cond	flags
0	EQ	Z=1
1	NE	Z=0
2	CS/HS	C=1
3	CC/LO	C=0
4	MI	N=1
5	PL	N=0
6	VS	V=1
7	VC	V=0
8	HI	C=1 && Z=0
9	LS	C=0 || Z=1
10	GE	N=V
11	LT	N!=V
12	GT	Z=0 &&N=V
13	LE	Z=0&&N!=V
14	AL	ANY
15		

status code

code	value	meaning
AOK	1	program executing normally
HLT	2	processor executed HALT
ADR	3	read/write of illegal adrs
INS	4	illegal instruction