	EECS 370: GREEN CARD FOR LEGv8											
Arithmetic Operations	Assembly code					Semantics				Comments		
add	ADD	Xd,	Xn,	Xm		X5 = X2 + X7				register-to-register		
add & set flags	ADDS	Xd,	Xn,	Xm			X5 = X2 + X7			flags NZVC		
add immediate	ADDI	Xd,	Xn,	#uimm12			X5 = X2 + #19			$0 \le 12$ bit unsigned $\le 4095$		
add immediate & set flags	ADDIS	Xd,	Xn,	#uimm12			X5 = X2 + #19			flags NZVC		
subtract	SUB	Xd.	Xn,	Xm			X5 = X2 - X7			register-to-register		
subtract & set flags	SUBS	Xd,	Xn,	Xm			X5 = X2 - X7		flags NZVC			
subtract immediate	SUBI	Xd,	Xn,	#uin	#uimm12		X5 = X2 - #20			$0 \le 12$ bit unsigned $\le 4095$		
subtract immediate & set flags	SUBIS	Xd,	Xn,	Xm		X5 = X2 - #20				flags NZVC		
Data Transfer Operations	As	sembly co	ode	S			antics Comment			ts		
load register	LDUR	DUR Xt,		imm9]		X2 = 1	M[X6, #18]	(	double word load into Xt from Xn + #simm9			
load signed word	LDURSW	Xt,	[Xn, #si	#simm91		X2 = M[X6, #18]		,	word load to lower 32b Xt from $Xn + #simm9$ ; sign extend upper 32b			
load half	LDURH	Xt,	[Xn, #si	, #simm9]		X2 = M[X6, #18]		1	$\frac{1}{2}$ word load to lower 16b Xt from Xn + #simm9; zero extend upper 48b			
load byte	LDURB	Xt,	[Xn, #si	[Xn, #simm9]		X2 = M[X6, #18]		ł	byte load to least 8b Xt from Xn + #simm9 zero extend upper 56b			
store register	STUR	Xt,	[Xn, #simm9]			M[X5, #12] = X4			double word store from Xt to Xn + #simm9			
store word	STURW	Xt,	[Xn, #simm9]			M[X5,	#12] = X4	,	word store from lower 32b of Xt to Xn + #simm9			
store half word	STURH	Xt,	[Xn, #si	[Xn, #simm9]			#12] = X4	1	<sup>1</sup> / <sub>2</sub> word load from lower 16b of Xt to Xn + #simm9			
store byte	STURB	Xt,	[Xn, #si	imm9]		M[X5,	[X5, #12] = X4 byte load		oyte load from	rom least 8b of Xt to Xn + #simm9		
offset		#simm9 =	$\#simm9 = -256 \ to \ +255$				-256			$256 \leq 9$ bits signed immediate $\leq +255$		
move wide with zero	MOVZ	Xd,	#uimm1	#uimm16, LSL N		X9 = 00N00		2 1 1	zeros out Xd then place a 16b (#uimm) into the first $(N = 0)$ /second $(N = 16)$ /third $(N = 32)$ /fourth $(N = 48)$ 16b slot of Xd			
move wide with keep	MOVK	IOVK Xd, #uimm16,				SL N $X9 = xxNxx$			place a 16b (#uimm) into the first (N = 0)/second (N = 16)/ third (N = 32)/fourth (N = 48) 16b slot of Xd, without changing the other values (x's)			
register aliases		X28 = SP; X29 = FP; X30 = LR; X31 = XZR										
Logical Operations	Assemb	oly code					Semantic	es		Using C operations of & $ ^{<} < > >$		
and	AND	Xd,	Xn,		Xm		X5 = X2 & X7		bit-wise AND			
and immediate	ANDI	Xd,	Xn,		#uimm12		X5 = X2 & #19		bit-wise AND with $0 \le 12$ bit unsigned $\le 4095$			
inclusive or	ORR	Xd,	Xn,		Xm		$X5 = X2 \mid X7$			bit-wise OR		
inclusive or immediate	ORRI	Xd,	, Xn,		#uimm12		X5 = X2   #11			bit-wise OR with $0 \le 12$ bit unsigned $\le 4095$		
exclusive or	EOR	Xd,	, Xn,		Xm		$X5 = X2^{X7}$			bit-wise EOR		
exclusive or immediate	EOR	Xd,	Kd, Xn,		#uimm12		$X5 = X2^{ + 57}$			bit-wise EOR with $0 \le 12$ bit unsigned $\le 4095$		
logical shift left	LSL	Xd,	Xd, Xn,		#uimm6		X1 = X2 << #			shift left by a constant $\leq 63$		
logical shift right	LSR	Xd,	Xn,	.n, #uimm6		X5 = X3 >> #		> #20	$20$ shift right by a constant $\leq 63$			
Unconditional branches	litional branches Asseml			bly code Semai				Also	Also known as Jumps			
branch	B	#simm26		goto BC   #1200		10.5		PC relative branch PC $\pm$ 26h offset: $-2^{25} \le \# cimm 26$				
braich	Б	#811111120		goio FC + #1200				$< 2^2 5-1$ : 4b instruc		tion		
branch to register	BR	Xt	Xt t		target in Xt			Xt con	tains a full 64	b address		
branch with link	BL	#simm26	#simm26		+ 4; PC + #1	11000	PC rela 16 milli X30 =		relative branch to PC + 26b offset; nillion instructions; ) = LR contains return from subroutine address			

Conditional	Asse	embly	y code	•			Se	mantics		Comments				
conditional branch	= 0		CBZ	Xt			#simm19		If $(X2 = = 0)$ goto PC + #99 if $\leq$			if $Xt = 0$ branch to PC + 19b offset: -2^18 4b instructions $\leq \#simm26 \leq 2^{18-1}$ 4b instructions		
conditional branch	!= 0	CBNZ			Xt		#simm19		If $(X2 != 0)$ go to PC + #89 if $\sum_{\leq 1}^{n}$			if $Xt = 0$ branch to PC + 19b offset: -2^18 4b instructions $\leq \#simm26 \leq 2^{18-1}$ 4b instructions		
branch conditionall	У	B.cond			l #simm19							if cond = true branch to PC +1 19b offset: $-2^{18}$ 4b instructions $\leq \#simm19 \leq 2^{18-1}$ 4b instructions		
Conditional	cases (con	(cond) Si			ied Ni	ımber	S	Unsigned N		umbers C		Comments		
=				B.EQ	B.EQ Z=			B.EQ	Z	=1	equal			
¥				B.NE	.NE Z=0			B.NE	Z	=0	not eq	lual		
<				B.LT	.LT N!			B.LO	С	=0	less th	ss than: or lower		
≤				B.LE	B.LE ~		0 & N=V)	B.LS	~	(Z = = 0 & N = V)	less th	nan or equal: or lower or same		
>				B.GT (Z		(Z=0	& N=V)	B.HI	(Z	Z = 0 & C = 1)	greate	er than: or higher		
2			B.GE	.GE N=V			B.HS	С	=1	great f	than or equal: or higher or same			
				B.MI		N = 1	B.PL	branch	on minu	is: branch on plus				
				B.VS	/S N=1		B.PL	branch on		verflow set: branch on overflow clear				
Notes on FLAGS NVZC				Set explicitly by arithmetic opera						ations with "S" in the mnemonic				
negative	N msb of result = 1									indicates a negative result if operands are two's complement				
oVerflow	V	$(carry out of msb) \otimes (carry out of msb-1) = 1$								indicates the result is an overflow if operands are two's complement				
zero	Z C	result = 0 carry out of msb = 1								indicates the result is all zeros				
carry	C							indicates a carry out				e msh of the result		
									indicates a carry out of the mot of the result					
Pseudoinstructions Assembly					code Semantics					Comments				
move reg-to-reg MOV			Xd Xn			Xd = Xn			text replacement for ORR Xd XZR Xn					
compare CM			CMP 2	P Xn, Xm		set flags NV	ZC		text replacement for	XZR, Xn, Xm				
compare immediate CMP				Xn, #uimm12			set flags NV	ZC		text replacement for	r SUBIS	XZR, Xm, #uimm12		

Copyright: Edgumn Rovert & University of Michigan EECS 370.