

AA-64 architecture

32bit mod R/M byte

												REX.R=1								
r8(/r) without REX				AL	CL	DL	BL	AH	CH	DH	BH	R8B	R9B	R10B	R11B	R12B	R13B	R14B	R15B	
r8(/r) with REX				AL	CL	DL	BL	SPL	BPL	SIL	DIL	R8W	R9W	R10W	R11W	R12W	R13W	R14W	R15W	
r16(/r)				AX	CX	DX	BX	SP	BP	SI	DI	R8D	R9D	R10D	R11D	R12D	R13D	R14D	R15D	
r32(/r)				EAX	ECX	EDX	EBX	ESP	EBP	ESI	EDI	R8	R9	R10	R11	R12	R13	R14	R15	
r64(/r)				RAX	RCX	RDY	RBX	RSP	RBP	RSI	RDI	MM0	MM1	MM2	MM3	MM4	MM5	MM6	MM7	
mm(/r)				MM0	MM1	MM2	MM3	MM4	MM5	MM6	MM7	XMM8	XMM9	XMM10	XMM11	XMM12	XMM13	XMM14	XMM15	
xmm(/r)				XMM0	XMM1	XMM2	XMM3	XMM4	XMM5	XMM6	XMM7	ES	CS	SS	DS	FS	GS	res.	res.	
sreg				ES	CS	SS	DS	FS	GS	res.	res.	CR8	CR9	CR10	CR11	CR12	CR13	CR14	CR15	
eee				CR0	CR1	CR2	CR3	CR4	CR5	CR6	CR7	DR8	DR9	DR10	DR11	DR12	DR13	DR14	DR15	
eee				DR0	DR1	DR2	DR3	DR4	DR5	DR6	DR7	0	1	2	3	4	5	6	7	
/digit (opcode)				0	1	2	3	4	5	6	7	000	001	010	011	100	101	110	111	
reg=				000	001	010	011	100	101	110	111	000	001	010	011	100	101	110	111	
effective address		effective address REX.B=1	mod	R/M	value of mod R/M byte (hex)								value of mod R/M byte (hex)							
[rAX]	[r8]	00	000	00	08	10	18	20	28	30	38	00	08	10	18	20	28	30	38	
[rCX]	[r9]		001	01	09	11	19	21	29	31	39	01	09	11	19	21	29	31	39	
[rDX]	[r10]		010	02	0A	12	1A	22	2A	32	3A	02	0A	12	1A	22	2A	32	3A	
[rBX]	[r11]		011	03	0B	13	1B	23	2B	33	3B	03	0B	13	1B	23	2B	33	3B	
[sib]	[sib]		100	04	0C	14	1C	24	2C	34	3C	04	0C	14	1C	24	2C	34	3C	
[*] #1	[*] #1		101	05	0D	15	1D	25	2D	35	3D	05	0D	15	1D	25	2D	35	3D	
[rSI]	[r14]		110	06	0E	16	1E	26	2E	36	3E	06	0E	16	1E	26	2E	36	3E	
[rDI]	[r15]		111	07	0F	17	1F	27	2F	37	3F	07	0F	17	1F	27	2F	37	3F	
[rAX+sbyte]	[r8+sbyte]	01	000	40	48	50	58	60	68	70	78	40	48	50	58	60	68	70	78	
[rCX+sbyte]	[r9+sbyte]		001	41	49	51	59	61	69	71	79	41	49	51	59	61	69	71	79	
[rDX+sbyte]	[r10+sbyte]		010	42	4A	52	5A	62	6A	72	7A	42	4A	52	5A	62	6A	72	7A	
[rBX+sbyte]	[r11+sbyte]		011	43	4B	53	5B	63	6B	73	7B	43	4B	53	5B	63	6B	73	7B	
[sib+sbyte]	[sib+sbyte]		100	44	4C	54	5C	64	6C	74	7C	44	4C	54	5C	64	6C	74	7C	
[rBP+sbyte]	[r13+sbyte]		101	45	4D	55	5D	65	6D	75	7D	45	4D	55	5D	65	6D	75	7D	
[rSI+sbyte]	[r14+sbyte]		110	46	4E	56	5E	66	6E	76	7E	46	4E	56	5E	66	6E	76	7E	
[rDI+sbyte]	[r15+sbyte]		111	47	4F	57	5F	67	6F	77	7F	47	4F	57	5F	67	6F	77	7F	
[rAX+sdword]	[r8+sdword]	10	000	80	88	90	98	A0	A8	B0	B8	80	88	90	98	A0	A8	B0	B8	
[rCX+sdword]	[r9+sdword]		001	81	89	91	99	A1	A9	B1	B9	81	89	91	99	A1	A9	B1	B9	
[rDX+sdword]	[r10+sdword]		010	82	8A	92	9A	A2	AA	B2	BA	82	8A	92	9A	A2	AA	B2	BA	
[rBX+sdword]	[r11+sdword]		011	83	8B	93	9B	A3	AB	B3	BB	83	8B	93	9B	A3	AB	B3	BB	
[sib+sdword]	[sib+sdword]		100	84	8C	94	9C	A4	AC	B4	BC	84	8C	94	9C	A4	AC	B4	BC	
[rBP+sdword]	[r13+sdword]		101	85	8D	95	9D	A5	AD	B5	BD	85	8D	95	9D	A5	AD	B5	BD	
[rSI+sdword]	[r14+sdword]		110	86	8E	96	9E	A6	AE	B6	BE	86	8E	96	9E	A6	AE	B6	BE	
[rDI+sdword]	[r15+sdword]		111	87	8F	97	9F	A7	AF	B7	BF	87	8F	97	9F	A7	AF	B7	BF	
AL/rAX/MM0/XMM0 CL/rCX/MM1/XMM1 DL/rDX/MM2/XMM2 BL/rBX/MM3/XMM3 AH/SPL/rSP/MM4/XMM4 CH/BPL/rBP/MM5/XMM5 DH/SIL/rSI/MM6/XMM6 BH/DIL/rDI/MM7/XMM7	R8 (B/W/D/Q)/MM0/XMM8	11																		
	R9 (B/W/D/Q)/MM1/XMM9																			
	R10 (B/W/D/Q)/MM2/XMM10		000	C0	C8	D0	D8	E0	E8	F0	F8	C0	C8	D0	D8	E0	E8	F0	F8	
	R11 (B/W/D/Q)/MM3/XMM11		001	C1	C9	D1	D9	E1	E9	F1	F9	C1	C9	D1	D9	E1	E9	F1	F9	
	R12 (B/W/D/Q)/MM4/XMM12		010	C2	CA	D2	DA	E2	EA	F2	FA	C2	CA	D2	DA	E2	EA	F2	FA	
	R13 (B/W/D/Q)/MM5/XMM13		011	C3	CB	D3	DB	E3	EB	F3	FB	C3	CB	D3	DB	E3	EB	F3	FB	
	R14 (B/W/D/Q)/MM6/XMM14		100	C4	CC	D4	DC	E4	EC	F4	FC	C4	CC	D4	DC	E4	EC	F4	FC	
	R15 (B/W/D/Q)/MM7/XMM15		101	C5	CD	D5	DD	E5	ED	F5	FD	C5	CD	D5	DD	E5	ED	F5	FD	
note		description																		
#1		if PM64 and adsize=64, then effective address = [RIP+sdword] if PM64 and adsize=32, then effective address = [EIP+sdword] else effective address = [sdword]																		

