AA-64 architecture
32bit SIB byte

											REX.B=1							
r32 base= base=				rAX 0 000	rCX 1 001	rDX 2 010	rBX 3 011	rSP 4 100	#1 5 101	rSI 6 110	rDI 7 111	r8 0 000	r9 1 001	r10 2 010	r11 3 011	r12 4 100	#2 5 101	r14 6 110	r15 7 111
scaled index	scaled index REX.X=1	S	index	value of SIB byte (hex)								value of SIB byte (hex)							
[rAX*1+base]	[r8*1+base]	00	000	00	01	02	03	04	05	06	07	00	01	02	03	04	05	06	07
[rCX*1+base]	[r9*1+base]		001	08	09	0A	0B	0C	0D	0E	0F	08	09	0A	0B	0C	0D	0E	0F
[rDX*1+base]	[r10*1+base]		010	10	11	12	13	14	15	16	17	10	11	12	13	14	15	16	17
[rBX*1+base]	[r11*1+base]		011	18	19	1A	1B	1C	1D	1E	1F	18	19	1A	1B	1C	1D	1E	1F
[none*1+base]	[r12*1+base]		100	20	21	22	23	24	25	26	27	20	21	22	23	24	25	26	27
[rBP*1+base]	[r13*1+base]		101	28	29	2A	2B	2C	2D	2E	2F	28	29	2A	2B	2C	2D	2E	2F
[rSI*1+base]	[r14*1+base]		110	30	31	32	33	34	35	36	37	30	31	32	33	34	35	36	37
[rDI*1+base]	[r15*1+base]		111	38	39	3A	3B	3C	3D	3E	3F	38	39	3A	3B	3C	3D	3E	3F
[rAX*2+base]	[r8*2+base]	01	000	40	41	42	43	44	45	46	47	40	41	42	43	44	45	46	47
[rCX*2+base]	[r9*2+base]		001	48	49	4A	4B	4C	4D	4E	4F	48	49	4A	4B	4C	4D	4E	4F
[rDX*2+base]	[r10*2+base]		010	50	51	52	53	54	55	56	57	50	51	52	53	54	55	56	57
[rBX*2+base]	[r11*2+base]		011	58	59	5A	5B	5C	5D	5E	5F	58	59	5A	5B	5C	5D	5E	5F
[none*2+base]	[r12*2+base]		100	60	61	62	63	64	65	66	67	60	61	62	63	64	65	66	67
[rBP*2+base]	[r13*2+base]		101	68	69	6A	6B	6C	6D	6E	6F	68	69	6A	6B	6C	6D	6E	6F
[rSI*2+base]	[r14*2+base]		110	70	71	72	73	74	75	76	77	70	71	72	73	74	75	76	77
[rDI*2+base]	[r15*2+base]		111	78	79	7A	7B	7C	7D	7E	7F	78	79	7A	7B	7C	7D	7E	7F
[rAX*4+base]	[r8*4+base]	10	000	80	81	82	83	84	85	86	87	80	81	82	83	84	85	86	87
[rCX*4+base]	[r9*4+base]		001	88	89	8A	8B	8C	8D	8E	8F	88	89	8A	8B	8C	8D	8E	8F
[rDX*4+base]	[r10*4+base]		010	90	91	92	93	94	95	96	97	90	91	92	93	94	95	96	97
[rBX*4+base]	[r11*4+base]		011	98	99	9A	9B	9C	9D	9E	9F	98	99	9A	9B	9C	9D	9E	9F
[none*4+base]	[r12*4+base]		100	A0	A1	A2	A3	A4	A5	A6	A7	A0	A1	A2	A3	A4	A5	A6	A7
[rBP*4+base]	[r13*4+base]		101	A8	A9	AA	AB	AC	AD	AE	AF	A8	A9	AA	AB	AC	AD	AE	AF
[rSI*4+base]	[r14*4+base]		110	B0	B1	B2	B3	B4	B5	B6	B7	B0	B1	B2	B3	B4	B5	B6	B7
[rDI*4+base]	[r15*4+base]		111	B8	B9	BA	BB	BC	BD	BE	BF	B8	B9	BA	BB	BC	BD	BE	BF
[rAX*8+base]	[r8*8+base]	11	000	C0	C1	C2	C3	C4	C5	C6	C7	C0	C1	C2	C3	C4	C5	C6	C7
[rCX*8+base]	[r9*8+base]		001	C8	C9	CA	CB	CC	CD	CE	CF	C8	C9	CA	CB	CC	CD	CE	CF
[rDX*8+base]	[r10*8+base]		010	D0	D1	D2	D3	D4	D5	D6	D7	D0	D1	D2	D3	D4	D5	D6	D7
[rBX*8+base]	[r11*8+base]		011	D8	D9	DA	DB	DC	DD	DE	DF	D8	D9	DA	DB	DC	DD	DE	DF
[none*8+base]	[r12*8+base]		100	E0	E1	E2	E3	E4	E5	E6	E7	E0	E1	E2	E3	E4	E5	E6	E7
[rBP*8+base]	[r13*8+base]		101	E8	E9	EA	EB	EC	ED	EE	EF	E8	E9	EA	EB	EC	ED	EE	EF
[rSI*8+base]	[r14*8+base]		110	F0	F1	F2	F3	F4	F5	F6	F7	F0	F1	F2	F3	F4	F5	F6	F7
[rDI*8+base]	[r15*8+base]		111	F8	F9	FA	FB	FC	FD	FE	FF	F8	F9	FA	FB	FC	FD	FE	FF

note	description
#1	if mod=00 then base=sdword if mod=01 then base=rBP+sbyte if mod=10 then base=rBP+sdword
#2	if mod=00 then base=sdword if mod=01 then base=r13+sbyte if mod=10 then base=r13+sdword