AA-64 architecture descriptors

available descriptor (P=0)																															
offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+4	available																0	DPL		S	TYPE		available									
+0	available																															

code (application) segment descriptor																															
offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+4	BASE (bit31..24)								G	D	L	AVL	LIMIT (bit19..16)				P	DPL		S=1	X=1	C	R	A	BASE (bit23..16)							
+0	BASE (bit15..0)																LIMIT (bit15..0)														

data (application) segment descriptor																															
offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+4	BASE (bit31..24)								G	B	r.	AVL	LIMIT (bit19..16)				P	DPL		S=1	X=0	E	W	A	BASE (bit23..16)							
+0	BASE (bit15..0)																LIMIT (bit15..0)														

LDT (system segment) descriptor

offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+12	reserved																			0	0	0	0	0	reserved							
+8	BASE (bit63..32)																															
+4	BASE (bit31..24)								G	r.	r.	AVL	LIMIT (bit19..16)				P	DPL			S=0	0	G=0	1	0	BASE (bit23..16)							
+0	BASE (bit15..0)																LIMIT (bit15..0)															

TSS (system segment) descriptor

offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+12	reserved																			0	0	0	0	0	reserved							
+8	BASE (bit63..32)																															
+4	BASE (bit31..24)								G	r.	r.	AVL	LIMIT (bit19..16)				P	DPL			S=0	D=1	G=0	B	V=1	BASE (bit23..16)							
+0	BASE (bit15..0)																LIMIT (bit15..0)															
note	Since the TR.ar.V bit is set to 0 during a processor RESET or INIT, it should act like a valid bit, and cause a #TS(0) exception on all implicit TSS accesses (stack switch, task switch, TSS32.IOPB, or TSS32.IRB). Most processors don't implement that behavior though.																															

call gate descriptor

offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

+12	reserved	0	0	0	0	0	reserved	
+8	OFFSET (bit63..32)							
+4	OFFSET (bit31..16)	P	DPL	S=0	D=1	G=1	00	res.	reserved
+0	PM64 CS selector	OFFSET (bit15..0)						

interrupt gate descriptor

offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+12	reserved																															
+8	OFFSET (bit63..32)																															
+4	OFFSET (bit31..16)																P	DPL			S=0	D=1	G=1	10	reserved						IST	
+0	PM64 CS selector																OFFSET (bit15..0)															

trap gate descriptor

offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+12	reserved																															
+8	OFFSET (bit63..32)																															
+4	OFFSET (bit31..16)																P	DPL			S=0	D=1	G=1	11	reserved						IST	

+0	PM64 CS selector	OFFSET (bit15..0)