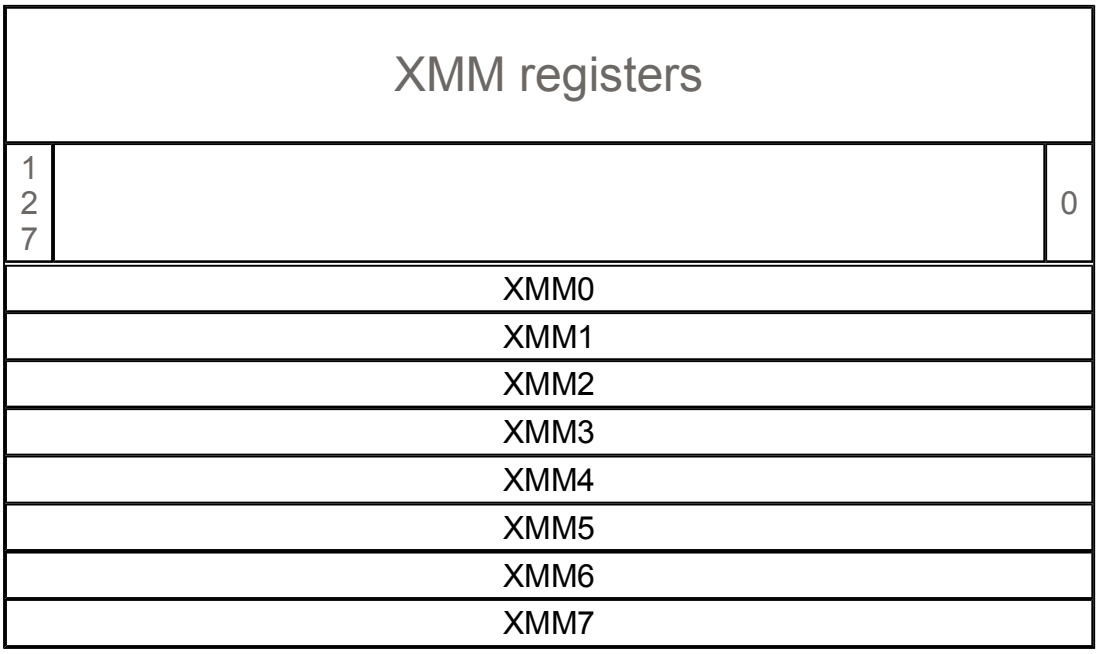
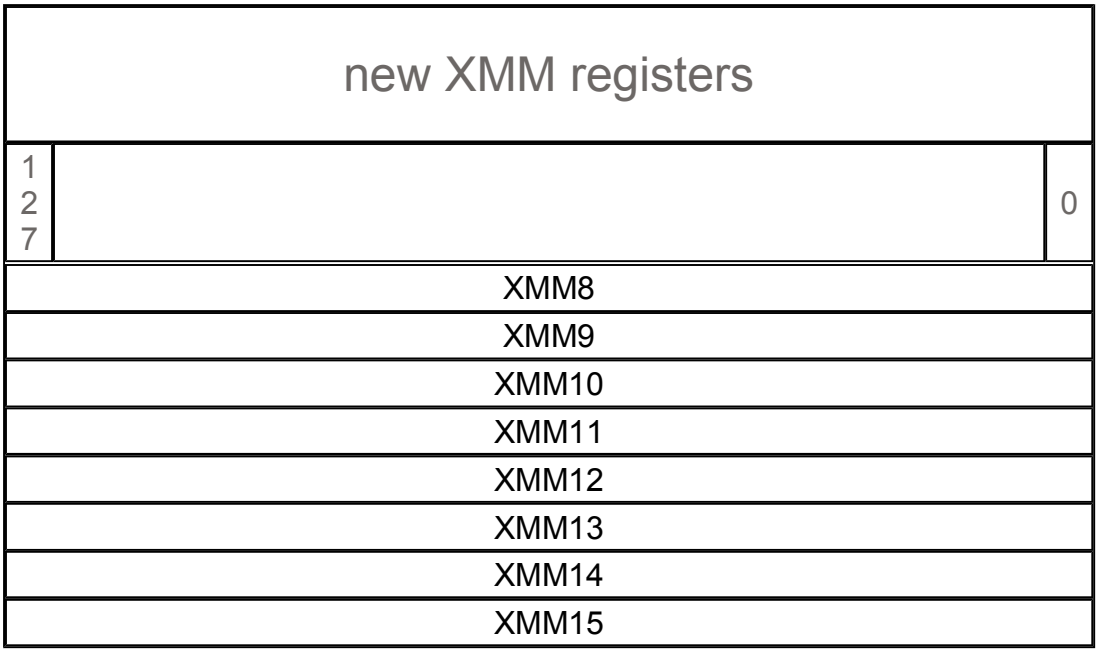
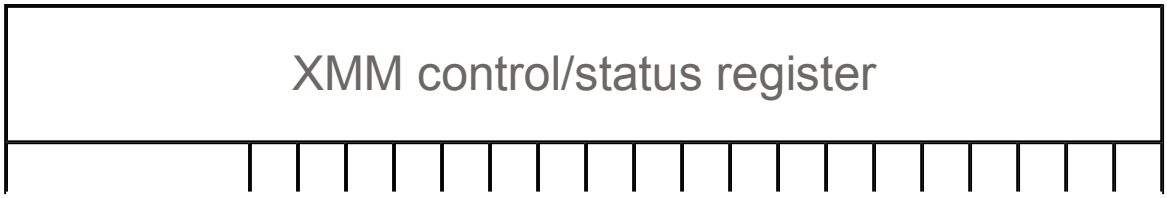
AA-64 architecture
XMM







register	31		16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MXCSR	res.			FZ	RC		PM	UM	OM	ZM	DM	IM	DAZ	PE	UE	OE	ZE	DE	IE

note: DAZ was introduced with SSE2. Check MXCSR_MASK (below) for more details.

LM FXSAVE and FXRSTOR format

here: if 16/32bit operand size

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	offset
reserved		FP_CS		FP_IP				FP_OPC		res.	FEMPTY	SW		CW		+0000h
MXCSR_MASK #1,#2				MXCSR #1				reserved		FP_DS		FP_DP				+0010h
reserved								ST(0) or MM0								+0020h
reserved								ST(1) or MM1								+0030h
reserved								ST(2) or MM2								+0040h
reserved								ST(3) or MM3								+0050h
reserved								ST(4) or MM4								+0060h
reserved								ST(5) or MM5								+0070h
reserved								ST(6) or MM6								+0080h
reserved								ST(7) or MM7								+0090h
XMM0 #1																+00A0h
XMM1 #1																+00B0h
XMM2 #1																+00C0h
XMM3 #1																+00D0h
XMM4 #1																+00E0h
XMM5 #1																+00F0h
XMM6 #1																+0100h
XMM7 #1																+0110h
XMM8 #1																+0120h
XMM9 #1																+0130h
XMM10 #1																+0140h
XMM11 #1																+0150h
XMM12 #1																+0160h

XMM13 #1	+0170h
XMM14 #1	+0180h
XMM15 #1	+0190h
reserved	+01A0h
reserved	+01B0h
reserved	+01C0h
reserved	+01D0h
reserved	+01E0h
reserved	+01F0h

note	description
#1	Whether or not these fields are saved/restored while CR4.OSFXSR=0 is implementation-specific.
#2	If this field is not written to by FXSAVE, then any MXCSR value that is to be loaded by FXRSTOR or LDMXCSR must be AND-masked with 0000_FFBFh first. Otherwise the value written by FXSAVE is to be used to AND-mask the MXCSR value before loading it with FXRSTOR or LDMXCSR.

64bit LM FXSAVE and FXRSTOR format

here: if 64bit operand size

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	offset
flat 64bit FP_IP								FP_OPC		res.	FEMPTY	SW		CW		+0000h
MXCSR_MASK #1,#2				MXCSR #1				flat 64bit FP_DP								+0010h
reserved								ST(0) or MM0								+0020h
reserved								ST(1) or MM1								+0030h
reserved								ST(2) or MM2								+0040h
reserved								ST(3) or MM3								+0050h
reserved								ST(4) or MM4								+0060h
reserved								ST(5) or MM5								+0070h
reserved								ST(6) or MM6								+0080h
reserved								ST(7) or MM7								+0090h
XMM0 #1																+00A0h
XMM1 #1																+00B0h
XMM2 #1																+00C0h
XMM3 #1																+00D0h
XMM4 #1																+00E0h

XMM5 #1	+00F0h
XMM6 #1	+0100h
XMM7 #1	+0110h
XMM8 #1	+0120h
XMM9 #1	+0130h
XMM10 #1	+0140h
XMM11 #1	+0150h
XMM12 #1	+0160h
XMM13 #1	+0170h
XMM14 #1	+0180h
XMM15 #1	+0190h
reserved	+01A0h
reserved	+01B0h
reserved	+01C0h
reserved	+01D0h
reserved	+01E0h
reserved	+01F0h

note	description
#1	Whether or not these fields are saved/restored while CR4.OSFXSR=0 is implementation-specific. The XMMn fields won't be saved/restored in CPL=0 PM64, if EFER.FFXSR is set to 1.
#2	If this field is not written to by FXSAVE, then any MXCSR value that is to be loaded by FXRSTOR or LDMXCSR must be AND-masked with 0000_FFBFh first. Otherwise the value written by FXSAVE is to be used to AND-mask the MXCSR value before loading it with FXRSTOR or LDMXCSR.