Memory Regions

Begin	End	Type	Comments	Size
0x00000000	0x00020000	R1	ATCM, GOOD=BAD	
0x00001455	0x00003764	code	SATA Read/Write Functions	
0x00011313	0x00013D09	code	SATA Functions	
0x0001334C	0x0001334C		ROM! This is set to 0 around the SATA code	
0x00800000	0x01000000	R2	BTCM, Contains Instructions (Bad-Cycle IP:=0x0081b810) [P11]	
0x00800000	0x00803400	bzero	Memset(0) in safe	
0x00800000	0x00801000	bzero		
0x00800000	0x00A28000	BTCM	BTCM for Mex1 Normal	
0x00800000	0x00801000	MEX2	MEX2 – bzero initialized	
0x00800C00	0x00800F20	NCQ-Ringbuffer	SATA requests arrive here	
0x0080106C	0x0080106C	Structure	Pointer to a 4 Buffers with each 76 Bytes, MEX3 related, pointed to by 0x81C648	
0x008010DC	0x008010DC	ENUM	Contains 1,2,4,6,8	
0x0080112C	0x0080113C	Array	Array with 3-4 DWORDS	
0x0080131C	0x0080131C	ENUM	Contains a value from 1-8 which gets switched	
0x00801320	0x00801320	structure	Structure that points to memory regions that are zeroized by MEX2 during initialisation	
0x008013F8	0x00827F00	bzero		
0x008013F8	0x008013F8	MEX2	MEX2 – bzero initialized	
0x00803400	0x00828000	bzero	Memset(0) in safe	
0x0080453C	0x0080453C	Pointer	Points to a buffer which contains a Function pointer at +88	
0x00804540	0x00804563	FunctionPointers	:=0x12E24 (likely an array of function pointers)	
0x00804564	0x00804564	Counter	Number of buffers to look for allocating 512 Blocks for SATA	
0x00804568	0x00804568	FAT	Allocation table 804568+i*12, 0=free 1=allocated	
0x0080471C	0x0080471C	Base Address	??? seems to be 0	
0x008049AC	0x008049AC	Base Address	In SAFE Mode: 0x800C00, in Normal mode: 0x8049C4	
0x00804E1C	0x00804E1C	Bool	In SAFE Mode: 0, in Normal mode: 0x804E34 SAFE-main()	
0x00805EBC	0x00805EBC	Base Address		
0x00808020	0x00808074		init:sub_fd82	
0x00808020	0x00808020	Base Address	0x20506000	
0x00808094	0x00808094	Pointer	0x81cb7c start address of the 64 entry command array, each entry is 32 bytes: idx, time, lba, sctcnt, cmdqr, c	
0x00808098	0x00808098	sizeof	0x34 sfr unit counter (sfr is a list of differently sized memory regions)	
0x00808098	0x00808098	Pointer	Pointer to the SFR List. The SFR List contains Base address of memory and Size of Memory region	
0x00808521	0x00808E4D	code	Security Firmware SATA Functions	
0x0080AA08			Base address for an array with 24 bytes per element, SATA related	

0x0080B3F0	0x0080B3F0	Base Address	Seems SATA related
0x0080BC08	0x0080BC08	Bool	Is queried by SAFE, but never written
0x0080BC17	0x0080BC17	Bool	Related to 0x20000000 PHY
0x0080C158	0x0080C158	bool	I2C Status is stored in this bit
0x0080C15C	0x0080C15C	Bool	Bool, Number of Bits MSB needs to be shifted left – WHICH VALUE
0x0080C160	0x0080C160	channel	Number of channels, I think this is 8 (in SAFE)
0x0080C160	0x0080C160	Base Address	Interesting base address
0x0080C420	0x0080C420	Pointer	Pointing to 0x82A00000
0x0080C420	0x0080C434		
0x0080C428	0x0080C428	nBlocks	Number of 0x1000 sized blocks
0x0080C42C	0x0080C42C	Size	=512
0x0080C42C	0x0080C42C	Blocksize	Blocksize of SATA related blocks
0x00816000	0x00954000	P11	
0x0081BDFC	0x0081C5F8	FTL	0x81BDFC+((4KB modulo 511) << 2)
0x0081C61C	0x0081C61C	Base Address	
0x0081C648	0x0081C648	Pointer	Points to the SATA NCQ buffer at 0x00800C00
0x0081C674	0x0081C678		:=0
0x0081C67C	0x0081C67C	Pointer	
0x0081C6A4	0x0081C6A4	MaxTemp	Maximum Temperature the SSD has had in its life
0x0081C6A8	0x0081C6A8	MinTemp	Minimum Temperature the SSD has had in its life
0x0081C6B8	0x0081C6B8	Counter	Interrupt Counter (statistics)
0x0081C6C0	0x0081C6C0	Counter	Meltdown Counter (how often was it too hot for the SSD)
0x0081C7BC	0x0081C7BC	Value	RESET_D2H_TIME in RDH_TIME_LOG
0x0081C7C4	0x0081C7C4	Pointer	Points to 4 Byte Command Array at 0x81C7CC
0x0081CAD8	0x0081CAD8	Pointer	Pointer to Firmware Version Array = 0x825A10
0x0081CB7C	0x0081CB7C	Array	Logfile Array for Commands, 32 Bytes
0x0081D390	0x0081D390	Bool	Is set to 1 by MEX1, seems to be 1 all the time
0x0081D39C	0x0081D39C	Number	:=26, Number of 8 Byte Elements in 0x81D3A0 Array
0x0081D3A0	0x0081D470	Array	Array with 8 Bytes per Element, 26 Elements
0x0081E500	0x0081E800	Array	64 Elements, 12 Bytes per Element, used by MEX1
0x008222C8	0x008222CB		
0x008232FC	0x008232FC	Size	Size of the SSD / Storage Capacity
0x00823304	0x00823304	Size	Maximum LBA Number: 488397167 / 0x1D1C596F
0x00823D04	0x00823D04	LBA	Seems to store a LBA

0x00824850	0x00824850	Bool	Seems to be always 0 in this firmware, perhaps it is for PCIe
0x00824C50	0x00824C50	Bool	
0x00824EBC	0x00824EBC	Pointer	Points to 0x20506000
0x00824EC4	0x00824EC4	?	0xffe0e005
0x00825A10	0x00825A10		Firmware Version should be here
0x00825BEC	0x00825BF3	structure	Structure initialized by MEX2, contains whether MAGIC is ok or not
0x00825C00	0x00825C00		points to 0x10020400, belongs to MEX3, this memory is likely individual to each core
0x00825C24	0x00825C2C	structure	Contains 2 important values for both MEX2 and MEX3 mainloop
0x00825C96	0x00825C96	Number	1..4, afterwards set to 0, related to MEX2
0x00826C00	0x00827C00	Stack	Stack for SVC Supervisor Mode for MEX3
0x00827C00	0x00827C80	Stack	Stack for FIQ Interrupt Mode for MEX3
0x00827C80	0x00827D80	Stack	Stack for IRQ Interrupt Mode for MEX3
0x00827D80	0x00827E00	Stack	Stack for UND Mode for MEX3
0x00827E00	0x00827F00	Stack	Stack for ABT Abort Mode for MEX3
0x10010000	0x10030000	R3	
0x10010000	0x10010040		SAFE starts an array with element size 16 here in dead code
0x1001004C	0x10010050		:=128
0x10010060	0x1001006F		MEX1 DMA PHY, 60=Status, 64=Size, 68=Source,6C=Target
0x10010070	0x1001007F		MEX2 DMA PHY, 70=Status, 74=Size, 78=Source,7C=Target
0x10010080	0x1001008F		MEX3 DMA PHY, 80=Status, 84=Size, 88=Source,8C=Target
0x10020000	0x1002007F		Array with 32 entries that gets initialized to 0
0x10020104	0x10020104		SAFE
0x10020190	0x10020190		:=0 in SAFE, likely to acknowledge UART
0x100201AC	0x100201AC		:=0 in SAFE
0x100201B4	0x100201B4		has a setter in SAFE
0x100201B8	0x100201B8		is always initialized by mex1
0x10020200	0x10020280		4 * v2 + 0x10020200
0x10020200	0x100203B8		belongs to MEX2
0x10020280	0x10020290		4 * (v2 >> 3) + 0x10020280
0x10020300	0x10020380		4 * v2 + 0x10020300
0x10020390	0x10020390		:=0
0x100203AC	0x100203AC		:=0
0x100203B0	0x100203B0		
0x100203B4	0x100203B4		:=-1

0x100203B8	0x100203B8		has a setter, sometimes set to -1 sometimes individual bits are set
0x10020400	0x100205B8		belongs to MEX3
0x10020800	0x10020800	IPC	Seems to be an unused IPC for Mex1
0x10020804	0x10020808	IPC	BTCM/SDRM/0000
0x10020808	0x10020808	IPC	BTCM/SDRM/0000
0x10020810	0x10020810	IPC	MEX3 sets this first to 0x88883164 and later to „BTCM“
0x10040000	0x10060000	R4	DEBUG ROM (CoreSight) + PHY
0x1004F010	0x1004F0AC	PHY	Initialized in normal 0x0000c658
0x1004F030	0x1004F030	PHY	Initialized in MEX1, related to
0x10050000	0x10050054		
0x10100000	0x10200000	R5	GOOD=BAD
0x20000000	0x20600000	R6	Data: Read from 0x203B000C
0x2000000C	0x20000124	SATAPHY	SATA PHY
0x20000010	0x20000010	Value	good:0x1010, bad:0x0
0x20000044	0x20000044	Value	???
0x20000048	0x20000048	LBA	LBA-LSB in SATA PHY
0x2000004C	0x2000004C	LBA	LBA-MSB in SATA PHY
0x20000054	0x20000054	Value	???
0x20000070	0x20000070	Value	???
0x20000094	0x20000094	PHY-Statusflag	
0x200000A4	0x200000A4	PHY-Statusflag	& 0x20000000 , &0x0400 is connection related
0x200000AC	0x200000AC	SATAPHY	Status register, 0x1000 is COMINIT seen
0x200000B0	0x200000B0	Magic	Magic value 0x044213D6 or 0x513E5
0x200000B4	0x200000B4	Magic	Magic value 0x801C4, writeable with |=0x90000
0x200000C0	0x200000C0	Address	Could be IRQ controller?
0x20000120	0x20000128	?	
0x20100000	0x201FFFFF		Might contain FLASH READ results
0x20100210	0x20100214		:=0x40800000
0x20100340	0x20100380	Array	16 Elements, 4 Bytes each
0x20100380	0x20100400	Array	16 Elements, 8 Bytes each
0x20101800	0x20101BFF	Array	256 Elements, 4 Bytes each
0x20101C00	0x20101FFF	Array	256 Elements, 4 Bytes each
0x20102010	0x20102010	PHY BIT	??? (SAFE)
0x20104000	0x20104003	PHY Bitfield	

0x20105004	0x20105004	PHY?	Seems SATA related
0x2010500C	0x2010500C	PHY?	Seems SATA related
0x20110000	0x201F0000	Copy	Identical copies of 0x20100000-0x2010FFFF, seems like an address line issue
0x20200000	0x202FFFFF	Array	2*8channels(?!?), with 64KByte each, points to 0x81BF7FF0
0x20205359	0x20205359		??? Is this a memory reference at all?
0x20300000	0x2030FFFF	channel	Channel 0, MEX2 is responsible
0x20300118	0x20300118	PHY Status	Status & 4 for Channel0
0x20300140	0x2030014C	channel	Seems to be a PHY register of the channel
0x20310000	0x2031FFFF	channel	Channel 1, MEX2 is responsible
0x20320000	0x2032FFFF	channel	Channel 2, MEX2 is responsible
0x20330000	0x2033FFFF	channel	Channel 3, MEX2 is responsible
0x20340000	0x20347FFF	Crypto/Channel	Crypto or Channel related
0x20340030	0x2034003C	EC	Wiriting 1 to 30 and 3C and sleep(10) starts 32-16 EC Curve loading
0x20340034	0x20340038	EC	Loading 32-16 EC Curve Data
0x2034003C	0x20340040	EC	Loading 32-32-12 EC Curve Data
0x20340048	0x2034004C	EC	Loading 32-4 EC Curve Data
0x20342000			Flash related
0x20344000	0x2034	channel	Elliptic Curves
0x20344034	0x20344034	EC	Writing a 1 and sleep(10) initializes 32-32-12 EC Curves
0x20344044	0x20344044	EC	Writing a 1 and sleep(10) initializes 32-4 EC Curves loading
0x2034A000	0x2034FFFF	Crypto/Channel	Crypto or Channel related
0x20380000	0x2038FFFF	Channel#0	MEX2
0x20380000	0x203C0000	channels	Array channelbase=((channel<<16)&0x3FFFF)+(channel>>2<<20)+0x20380000
0x20390000	0x2039FFFF	Channel#1	MEX2
0x203A0000	0x203AFFFF	Channel#2	MEX2
0x203B0000	0x203B0000	Channel#3	MEX2
0x203C0000	0x203F0000	channel	Something channel related, not sure about the size
0x203C005C	0x203C005C		Looks like a target address
0x203C0400	0x203C0400	channel	more specific base address
0x203C3C00	0x203C3E00	channel	Channel 0, Destination 15
0x20400000	0x2040FFFF	channel	Channel 4, MEX3 is responsible
0x20410000	0x2041FFFF	channel	Channel 5, MEX3 is responsible
0x20420000	0x2042FFFF	channel	Channel 6, MEX3 is responsible
0x20430000	0x2043FFFF	channel	Channel 7, MEX3 is responsible

0x20480000	0x2048FFFF	Channel#4	MEX3
0x20490000	0x2049FFFF	Channel#5	MEX3
0x204A0000	0x204AFFFF	Channel#6	MEX3
0x204B0000	0x204BFFFF	Channel#7	MEX3
0x204C0000	0x204CFFFF	Channel#4	MEX3, seperated in 1024 Bytes per Block
0x204D0000	0x204DFFFF	Channel#5	MEX3, seperated in 1024 Bytes per Block
0x204E0000	0x204EFFFF	Channel#6	MEX3, seperated in 1024 Bytes per Block
0x204F0000	0x204FFFFF	Channel#7	MEX3, seperated in 1024 Bytes per Block
0x20500000	0x205F0000		Various PHYs
0x20500004	0x20500030	PHY	
0x20501000	0x20501020	I2C	I2C PHY, seems to be connected to UART too, perhaps even Flash?!?
0x20501038	0x20501038		Contains something interesting in the upper 16 bits, and a bitfield
0x20501204	0x20501204	TIME	UPTIME in negative milliseconds → overflows after 49.71 days – BUG?
0x20502000	0x20502014	PHY	sub_AF9C
0x2050200C	0x2050200C	CORE	This seems like waking up a CPU core
0x20503000	0x20503020	UART	UART PHY
0x20503014	0x20503014	UART	UART Write Byte
0x20503018	0x20503018	UART	UART Read Byte
0x20503024	0x20503024	UART	
0x20504000	0x20504FFF	PHY	Some PHY that is not used in SAFE
0x20504000	0x20504000	PHYSTATUS	Status bit 0x10 of a PHY
0x20506000	0x20506FFF	PHY	is initialized in SAFE
0x20506024	0x20506024	Counter	Seems to be some strange counter
0x20506044	0x20506044	TIME	UPTIME in negative 4KHz integer → overflows after 12.4275 days
0x2050F024	0x2050F024	Bitfield	Some bits need to be set for MEX2 to start correctly
0x2050F038	0x2050F038	PHY – switch	Related to 0x20500000, values 1,2,4,6,8 or other
0x2050F03C	0x2050F03C	Bitfield	Some bits get cleared
0x40000000	0x40020000	P21	
0x40000000	0x43000000	R7	(starts with GOOD=BAD) [P21+P22+P31+P32+P41+P42]
0x40808000	0x40810000	P22	
0x40815E00	0x40815E00	channel	0x1800*prev*1_2 +0x40815E00
0x41000000	0x41020000	P31	
0x41800000	0x4180FFFF		MEX2 related, initialized by MEX1
0x41801000	0x41805000	P32	

0x41827FFC	0x41827FFC	PHY	0x46878568, 0 This could be waking/sleeping MEX2
0x42000000	0x42020000	P41	
0x42800000	0x4280FFFF		MEX3 related, initialized by MEX1
0x42801000	0x42805000	P42	
0x42827FFC	0x42827FFC	PHY	0x46878568, 0 This could be waking/sleeping MEX3
0x44000000	0x47000000	R8	(starts with GOOD=BAD)
0x44803400	0x4480344C		Used by MEX3
0x48000000	0x4B000000	R9	(starts with GOOD=BAD)
0x80000000	0xA0000000	R10	RAM - [P23+P33+P43] - Samsung 512 MB LPDDR2 SDRAM, but mostly readonly
0x80000024	0x80000024	Magic	0x29135201, this is set when SA is loaded correctly from flash
0x80000028	0x80000028	Magic	0x66224266, this is set when SA is loaded correctly from flash
0x8000002C	0x8000002C	Magic	0x85661943
0x80000030	0x80000030	Magic	0xA0B0C0D
0x80000200	0x80030200	P23	
0x80000CB5	0x80001613	code	SATA NCQ Functions
0x800021A0	0x800021A0	code	Jumpout
0x800021A0	0x800021A0	code	
0x800021fc	0x800021fc	code	sata_45h_WRITE_UNCORRECTABLE_EXT
0x800021FD	0x80002D7D	code	SATA Functions
0x80002d7c	0x80002d7c	code	sata_83h_SECRET_COMMAND
0x80005793	0x80005793	code	SATA Defect List
0x8000b25c	0x8000b25c	code	sata_5Bh_TCG_TRUSTED_NONDAT_SEND_RECEIVE
0x8000B25D	0x8000B25D	code	SATA TCG Functions
0x8000C74C	0x8000C74C	code	Jumpout
0x8000C74C	0x8000C74C	code	
0x8000DCE3	0x8000DCE3	code	ERROR HANDLER – interesting !!!
0x800300F0	0x80030288		Array with Various addresses for Dumping, including channel base addresses
0x80030290	0x80030338		Registers are stored here for Dumping
0x800302C8	0x800302EC		Error handler area
0x800302F0	0x80030310		Buffer to hold Status Registers of the Processor for Errorhandler
0x80030341	0x80030341	Bool	This seems to be a requirement for error logging
0x80030348	0x80030348		Can be set to ‚INIT‘
0x80030348	0x80030348	Pointer	Pointer to Buffer with CPU Core Ids

0x80030354	0x80030354	Pointer	Mother of all Stack Regions = 0x803A0200
0x80040000	0x80064000	P33	
0x800413B0	0x800413B0	code	Function called by mex2
0x800413E4	0x800413E4	code	Function called by mex2
0x8004163A	0x8004163A	code	Function called by mex2
0x80051EDC	0x80051EDC	code	Function called by mex2
0x80051F80	0x80051F80	code	Function called by mex2
0x80051FE8	0x80051FE8	code	Function called by mex2
0x80052054	0x80052054	code	Function called by mex2
0x80052054	0x80052054	code	Function called by mex2
0x800520CC	0x800520CC	code	Function called by mex2
0x80052138	0x80052138	code	Function called by mex2
0x800521B0	0x800521B0	code	Function called by mex2
0x800521B0	0x800521B0	code	Function called by mex2
0x800524E4	0x800524E4	code	Function called by mex2
0x800527C4	0x800527C4	code	Function called by mex2
0x800566DC	0x800566DC	code	Function called by mex2
0x80057964	0x80057964	code	Function called by mex2
0x80061858	0x80060000	bzero	
0x80061858	0x80060000	MEX2	MEX2 – bzero initialized BUG!
0x80080000	0x800A4000	P43	
0x80080180	0x80081544	code	DEBUG Functions in SA, report any JMP to those addresses!
0x8008051C	0x8008051C		CORE→LR Register
0x80080520	0x80080520		CORE→PC Register
0x8008156E	0x8008156E	Error Handler	Relevant Error Handler for MEX3
0x800A1AC1	0x800A1AC1	Bool	Relevant for the Error handler
0x800A1AC1	0x800A1AC1	Error Handler	Relevant Error Handler for MEX3
0x800A1C68	0x800AED89	Structures	MEX3 related structures
0x80100000	0x801FFFFF		perhaps channel related
0x80100200	0x80100300	Bitfield	MEX1 – related to 0x81C00000 – length unknown
0x803A0200	0x803AFFFF		Related to stack regions
0x80421200	0x80521200	RAM	4K Block addresses for Delegation, likely at least 256
0x80427200	0x8140E200	RAM	Normal Flash-RAM area for MEX2
0x80445200	0x80BCC200		

0x80921800	0x80A21800	RAM	SATA-SAFE-SATA-BUFFER with 256 4K blocks, copied from SAFE-RAM-BUFFER
0x81801A00	0x81881A00	LOG	Logging area for 1024x512-Byte blocks, used by MEX1. Structure: 4 Bytes ID, 4 Bytes Uptime, 504 Bytes Dat
0x81881A00	0x818A1A00	Memcopy	memcpyDMA(0x45000000,0x81881A00,0x20000)
0x818A1A00	0x818C9A00	Memcopy	memcpyDMA(0x45800000,0x818A1A00,0x28000)
0x818C9A00	0x818E9A00	Memcopy	memcpyDMA(0x4A000000,0x81881A00,0x20000)
0x818E9A00	0x81911A00	Memcopy	memcpyDMA(0x4A800000,0x818E9A00,0x28000)
0x81911A00	0x81931A00	Memcopy	memcpyDMA(0x40000000,0x81911A00,0x20000)
0x81959A00	0x8195F200	Memcopy	memcpystrange(203C0000,81959A00,0x5800)
0x81969A00	0x8196F200	Memcopy	memcpystrange(203D0000,81969A00,0x5800)
0x81979A00	0x8197F200	Memcopy	memcpystrange(203E0000,81979A00,0x5800)
0x81989A00	0x8198F200	Memcopy	memcpystrange(203F0000,81989A00,0x5800)
0x819D9A00	0x819D9A04	Magic	Strt
0x819D9A40	0x81ACDC80	Stringbuffer	Exception-XML Stringbuffer
0x81BF7FF0	0x81BF7FF0		Pointed to by P21→subD636 from 0x202X0010, and by P21→BCC8
0x81C00000	0x81C00000		Pointed to by P21→B5A0
0x81C00000	0x81C00100	BitfieldStruct	MEX1 – related to 0x80100200 – length unknown
0x82A00000	0x9F800000		Init by safe_11E00
0x82C93000	0x88B3A000	LBA32K	4 RAM bases addresses found in MEX2, likely for the 4 channels
0x844ff000	0x845a2ac0	PhysicalBlockTable	MEX2: PBN*80 – Contains ReadCount and EraseCount
0x845a2ac0	0x84646580	PhysicalBlockTable	MEX2: PBN*80 – Contains ReadCount and EraseCount
0x849e81a0	0x849e9000	Bitfield	FTL Bitfield (at least MEX2) for MB2432
0x85833000	0x9F7FE000	RAM	SATA-SAFE-RAM-BUFFER
0x92eff000	0x92fa2ac0	PhysicalBlockTable	MEX3: PBN*80 – Contains ReadCount and EraseCount
0x92fa2ac0	0x93046580	PhysicalBlockTable	MEX3: PBN*80 – Contains ReadCount and EraseCount

cmdqw, cmd, tag, cmd_step

ta

Multipliers

MEX	times	multiplier	Startaddress	Pointer	Explanation	Relationship
		4		8010B0		
		4		801060		
		4		80112C		
		4		801078		
		4		801078		
		4		824930		
		4		80065884		
		4		800653F4		
	4	4		80061B40		
	16	4		80061B40		
	32	4		0x10020200		
	32	4		0x10020300		
	4	4		0x10020280		
	4	4		0x10020380		
	19	4		80097BF0		
	4	8		80106C	channels	
		8		801088		
		8		801176		
		8		80065354		
		8		800655D4		
		8		800688A4		
		8		80064A34		
		8		801172		
	15	8		801175	Byte	
		16		81C00000	512 Byte Blöcke: v8013FC = 0x81C00000 + 16 * (blockaddres	
	4	20		8256E0		
		20		800647A4		
		24		1B258		
	3	24		800815CC		
	2	26				
1	64	32	808094→81CB7C		COMMANDOs – MEX1 0x0081CB7C, Counter: 81CB54, related to 81CB4C	+0 INDEX +4
	16	44		8008055C	Exceptions DUMP	

		52			
		52,2			
		56		80109C	
	4	60,4			REL2
	256Byte	64,4		80061B40	80061B40+ (v0_3 << 6) + 4 * v0_15++;
		72			
23	4	76	801520	80106C	Shared by MEX2 and MEX3 , v80106C+76*LBA8Kmod4
		76		8010AC	
	4	76			
		80	801650	8010A8	
	v8010DC	80	20BC	8010E0	
		84		80065734	
		92		824CE4	
		92		824CE0	
		92		824C90	
		92		800659D4	
		100		80065444	
		127			Exceptions DUMP
	4	140			
		144		800647F4	
		180		8010F0	
		198		1A548	
		200		824970	
	4	244,36	1C880	8010A4, 8010A8	
		320			
	4	323			
		452		8010C0	
	4	452	80063B50	8010C8	Channels, 8010C8→800A3B68 on mex3
	4	516	823168	801098	Channels
		520		8010D0	
	4	528			
		528			
		560		80064A94	
		568		824E00	

		736,4		801094		
		768,67				REL1
		1040		8257E4		
		1200,2				
	4	1532		8010A0		
2	4	2052	821158	801070	Channel related: 821158,82195C,8822160	
		2304		80108C		
		2668		8010DC		
		2904		80065B44		
		3760			x=v/3760, y=v%3760	
	110	4096		8010C8→[452*x].448		
		8192			<<13, 8192=0x2000	
		12288			**a1+12288**(a2/3760), 12288=0x3000	
	4+4*v8010DC	18872			49B8	
		33600			0x8340	
		75488			0x126E0	
		<<16,<<10				REL1
	4	1024,508,4		8008055C		
		112,14,2				
	4	120,40				REL2
		120,60				
		140,16,4				
		200,10		824970		
		2668,16,4		801090		
		568,284		824E00		
		60,20				
	4	72,16				
		92,4		824C90		
	3				CPU cores (MEX1-MEX3)	
	4				Flash Channels	
	8				Flash Channels	
	16				Number of Hashtable Entries	
	16				Something each of the 8 flash channels has	
	32				NCQ Buffers	

33				NCQ Buffers	
64				SATA response buffers	
80				per Flash channel	
	512			LBA Block-Size	
	8192			1 Page = Minimum Read Size	2^13
	8832			1 Page = (8K + 640)Bytes	
	4152			Blocks per Device , 0x81B80	
	256			1 Block = 256 Pages = 2260992 Bytes = Min Er	2^8
	2260992			Bytes per Block = 0x228000	
	17664			2 Pages = 16KiB	
	531456			Pages per Device	
	8304			Valid Blocks per Device = 0x2070	
	16608			Valid Blocks per Device = 0x40E0	
	2097152			Data-Only size of a Block = 2 MiB	2^21
	4096			LBA blocks per Block	2^12
	250.059.349.504			Bytes per SSD	
	119238			Blocks per SSD	
	131072			Maximum SATA Request Size = 128 KiB	
	16			128 KiB **SATA requests per** 2MiB **Block**	
26	8		81D3A0		
144	20		81E80C		
33	16	800C00	81C648	SATA NCQ Buffer	
64	4		0x81C7CC		
	4294967			v2=v824E8c/1000+4294967*v4	
!=255	32	(40)801000	81C67C		REL3
!=255	4	20101800	81C680		REL3
	488397167			Number of 512 Byte Blocks available on the SSD	
511	4		81BDFC		

ss >> 9)

TIME +8 LBA +12 SCTCNT +16 CMDQR +18 CMDQW +20 CMD +24 TAG +28 CMD_STEP

```
v7 = ((a1 << 16) & 0x3FFFF) + (a2 << 10) + dword_80097BE0;

((a1 << 16) & 0x3FFFF) + (a2 << 10) + dword_80097BE0;

*(_BYTE *)(v10 + 44 * ((v28 + i) % 0x10) + 0xE1F8)
v10 = *(_DWORD *)(*(_DWORD *)(off_8008051C + 0x40) + 4 * (127 * v8 + (v9 << 8)) + 172);
```

Block calculations

GB	Bytes	512-Blocks	billion 512-Blocks	32-Bit Bytes	32-Bit GB	8K Pages	8K Bytes	8K MB	Blocks
128	137438953472	268435456	0,0625	1073741824	1	16777216	67108864	64	65536
256	274877906944	536870912	0,125	2147483648	2	33554432	134217728	128	131072
512	549755813888	1073741824	0,25	4294967296	4	67108864	268435456	256	262144
768	824633720832	1610612736	0,375	6442450944	6	100663296	402653184	384	393216
1024	1099511627776	2147483648	0,5	8589934592	8	134217728	536870912	512	524288
232,9	250059349504	488397167	0,1	1953588668	1,8	30524822,9	122099291,8	116,4	119238
	250059349504								
	QW=QueueWrite		8380	33520					
	QR=QueueRead								

Block#	6220336		597294	6220336	
Block# HEX	**0x005EEA30**	512-Byte LBA	**91D2E**	**5EEA30**	**0**
	0x000BDD46	4KB Alignment	**123A5**	**BDD46**	**0**
	0x000005F1	511 4KB-Blocks	**92**	**5F1**	**0**
	0x0005EEA3	8KB Alignment	**91D2**	**5EEA3**	**0**
	0x00000137	4KB modulo 511	**37**	**137**	**0**
	0x00000000	8KB bit (8KB & 1)	**FALSCH**	**WAHR**	**FALSCH**
		32KB	**2474**	**17BA8**	**0**
		8KBmod4	**2**	**3**	**0**
		32KBmod128	**74**	**28**	**0**
		32KBdiv3760			
		32KBmod3760			
	0x00824F3C	Base address with 8KB bit - [0x00824FF0]:=>00824f3c			
	0x0081C2D8	Base address with 4KBmod511<<2	**81BED8**	**81C2D8**	**81BDFC**
	0x0081BDFC	Base address without			
	0x00824F40	Base address with 8KB bit			
	8502780				

		8380
Mex1→Mex3 handover:		256
LBA8k	0x8002F0	2145280

Block calculations

16384	
35148267520	33520
	268160

Block calculations

4 Bytes per Block	KiB	32KB Blocks	32KB BlocksX	32div3760
262144	256	4194304	400000	1116
524288	512	8388608	800000	2231
1048576	1024	16777216	1000000	4462
1572864	1536	25165824	1800000	6693
2097152	2048	33554432	2000000	8924
476950	466	7631206	747165	2030

0x00007cdc 0x00007cde LDR
0x00007cde 0x00007ce0 STR

0x000021be 0x000021c0 LDR
0x000021c0 0x000021c2 STR

0x00000798 0x0000079c LDRD
0x0000079c 0x0000079e MOVS
0x0000079e 0x000007a0 LDRB

AddCMD:
0x000193f0 0x000193f2 MOV
…
0x00019420 0x00019424 STRD

Is the Request within the device or beyond the e
0x00001354 0x00001358 LDRD
0x00001358 0x0000135a LDR
0x0000135a 0x0000135c MOVS
0x0000135c 0x0000135e LDRB
0x0000135e 0x00001360 ADD
0x00001360 0x00001362 SUBS
0x00001362 0x00001364 CBZ
0x00001364 0x00001366 CMP
0x00001366 0x00001368 BCC
0x00001368 0x0000136a CMP
0x0000136a 0x0000136c BCC
0x0000136c 0x0000136e LDRB
0x0000136e 0x00001370 CBNZ
0x00001370 0x0000137e B
0x0000137e 0x00001380 MOV

HIER WEITERMACHEN

0x00002700	0x00002704	LDRD
…		
0x000027ba	0x000027be	LDRD
0x000027be	0x000027c0	MOVS
0x000027c0	0x000027c2	LDR
0x000027c2	0x000027c4	LDR
0x000027c4	0x000027c6	CMP
0x000027c6	0x0000282e	BNE
0x0000282e	0x00002830	LDR
0x00002830	0x00002832	ADDS
0x00002832	0x00002834	CMP
0x00002834	0x000027c2	BCC
0x000027c2	0x000027c4	LDR
0x000027c4	0x000027c6	CMP
0x000027c6	0x0000282e	BNE
0x0000282e	0x00002830	LDR
0x00002830	0x00002832	ADDS
0x00002832	0x00002834	CMP
0x00002834	0x000027c2	BCC
0x000027c2	0x000027c4	LDR
0x000027c4	0x000027c6	CMP
0x000027c6	0x0000282e	BNE
0x0000282e	0x00002830	LDR
0x00002830	0x00002832	ADDS
0x00002832	0x00002834	CMP
0x00002834	0x000027c2	BCC
0x000027c2	0x000027c4	LDR
0x000027c4	0x000027c6	CMP
0x000027c6	0x0000282e	BNE
0x0000282e	0x00002830	LDR
0x00002830	0x00002832	ADDS
0x00002832	0x00002834	CMP
0x00002834	0x00002836	BCC
0x00002836	0x00002838	LDR

0x00002838 0x0000283c ADD.W
0x0000283c 0x0000283e STRB
0x0000283e 0x00002842 STRD
0x00002842 0x00002844 STR

0x000028ca 0x000028ce LDRD
0x000028ce 0x000028d0 STR
0x000028d0 0x000028d2 LDR

0x00002984 0x00002986 LDR
0x00002986 0x0000298a AND

0x000029ae 0x000029b2 STRD
0x000029b2 0x000029b4 LDR
0x000029b4 0x000029b6 LSRS

0x00009e3a 0x00009e3c LSRS
0x00009e3c 0x00009e40 STRH.W
0x00009e40 0x00009e44 CMP.W
0x00009e44 0x00009e46 STR
0x00009e46 0x00009e4a AND
0x00009e4a 0x00009e4c STRH
0x00009e4c 0x00009e50 ADD.W
0x00009e50 0x00009e52 LDR

The following is a memcpy function:

0x0001e34c 0x0001e350 LDM
0x0001e350 0x0001e354 SUBS
0x0001e354 0x0001e358 STM

0x0001e358 0x0001e35c LDM

0x00002be2	0x00002be4	LDR
0x00002be4	0x00002be8	ADD.W
0x00002be8	0x00002bea	STR
0x00002bea	0x00002bec	LDR
0x00002bec	0x00002bf0	SUB.W

r2, [r0, #0x4] r2:0x0001FD10=>0x005EEA30 r0=0x00800C30 [0x00800C34]:=>005eea30 C
r2, [r4, #0x4] r2=0x005EEA30 r4=0x0081FC74 [0x0081FC78]:=>005eea30 C

r3, [r0, #0x18] r3:0x00827C44=>0x005EEA30 r0=0x0081FC60 [0x0081FC78]:=>005eea30 C
r3, [r0, #0x30] r3=0x005EEA30 r0=0x0081FC60 [0x0081FC90]:=>005eea30 C

r0, r2, [r4, #0] ; 00 r0:0x00000008=>0x00800C30 r2:0x00000001=>0x005EEA30 r4=0x0081FC74 [0x0081FC74]:=>00800c30 C
r4, #0x03 r4:0x0081FC74=>0x00000003 cpsr:0x60000133=>0x20000133 C
r1, [r0, #0x3] r1:0x00800C30=>0x00000060 r0=0x00800C30 [0x00800C33]:=>5eea3060 C

r7, r2 r7:0x00000004=>0x005EEA30 r2=0x005EEA30 C

r7, r8, [r4, #8] ; 0x08 r7=0x005EEA30 r8=0x00000008 r4=0x0081CCBC [0x0081CCC4]:=>005eea30 C

end?
r4, r3, [r0, #4] ; 0x04 r4:0x0081FC60=>**0x005EEA30** r3=0x00000008 r0=0x0081FC74 [0x0081FC78]:=>**005eea30** C
r2, [r2, #0xc] r2:0x008232F8=>0x1D1C596F [0x00823304]:=>1d1c596f C
r1, #0x01 r1:0x0000261F=>0x00000001 C
r5, [r0, #0xf] r5=0x00000001 r0=0x0081FC74 [0x0081FC83]:=>00000001 C
r3, r4 r3:0x00000008=>0x005EEA38 r4=0x005EEA30 C
r3, r3, #1 r3:0x005EEA38=>0x005EEA37 cpsr:0x00000133=>0x20000133 C
r5, 0x00001372 r5=0x00000001 C
r2, r3 r2=0x1D1C596F r3=0x005EEA37 C
0x0000137c C
r2, r4 r2=0x1D1C596F r4=0x005EEA30 C
0x0000137c C
r0, [r0, #0x10] r0:0x0081FC74=>0x00000000 [0x0081FC84]:=>00000000 C
r0, 0x0000137c r0=0x00000000 C
0x0000137e C
r0, r1 r0:0x00000000=>0x00000001 r1=0x00000001

r2, r3, [r7, #4] ; 0x04 r2:0xBE6B6CED=>0x005EEA30 r3:0x00000001=>0x00000008 r7=0x0081FC74 [0x0081FC78]:=>005eea30 C

r12, r14, [r7, #4] ; 0x04 r12:0x00822F68=>0x005EEA30 r14=0x9997FC4B r7=0x0081FC74 [0x0081FC78]:=>005eea30 C
r2, #00 r2:0x005EEA30=>0x00000000 cpsr:0x20000133=>0x60000133 C
r0, [r1, #0x58] r0:0x00000000=>0x00825044 r1=0x00825014 [0x0082506C]:=>00825044 C
r5, [r0, #0x4] r5:0xFFFFFFFF=>0x00001008 r0=0x00825044 [0x00825048]:=>00001008 C
r5, r12 r5=0x00001008 r12=0x005EEA30 cpsr:0x60000133=>0x80000133 C
0x0000282e C
r0, [r0, #0x10] r0:0x00825044=>0x0082501C [0x00825054]:=>0082501c C
r2, r2, #1 r2:0x00000000=>0x00000001 cpsr:0x80000133=>0x00000133 C
r2, #0x04 r2=0x00000001 cpsr:0x00000133=>0x80000133 C
0x000027c2 C
r5, [r0, #0x4] r5:0x00001008=>0x00000208 r0=0x0082501C [0x00825020]:=>00000208 C
r5, r12 r5=0x00000208 r12=0x005EEA30 C
0x0000282e C
r0, [r0, #0x10] r0:0x0082501C=>0x00825030 [0x0082502C]:=>00825030 C
r2, r2, #1 r2:0x00000001=>0x00000002 cpsr:0x80000133=>0x00000133 C
r2, #0x04 r2=0x00000002 cpsr:0x00000133=>0x80000133 C
0x000027c2 C
r5, [r0, #0x4] r5:0x00000208=>0x00000400 r0=0x00825030 [0x00825034]:=>00000400 C
r5, r12 r5=0x00000400 r12=0x005EEA30 C
0x0000282e C
r0, [r0, #0x10] r0:0x00825030=>0x00825058 [0x00825040]:=>00825058 C
r2, r2, #1 r2:0x00000002=>0x00000003 cpsr:0x80000133=>0x00000133 C
r2, #0x04 r2=0x00000003 cpsr:0x00000133=>0x80000133 C
0x000027c2 C
r5, [r0, #0x4] r5:0x00000400=>0x00000078 r0=0x00825058 [0x0082505C]:=>00000078 C
r5, r12 r5=0x00000078 r12=0x005EEA30 C
0x0000282e C
r0, [r0, #0x10] r0:0x00825058=>0x00825044 [0x00825068]:=>00825044 C
r2, r2, #1 r2:0x00000003=>0x00000004 cpsr:0x80000133=>0x00000133 C
r2, #0x04 r2=0x00000004 cpsr:0x00000133=>0x60000133 C
0x000027c2 C
r0, [r0, #0xc] r0:0x00825044=>0x00825058 [0x00825050]:=>00825058 C

r2, r12, r14 r2:0x00000004=>0x005EEA38 r12=0x005EEA30 r14=0x9997FC4B C
r3, [r0, #0x1] r3=0x00000001 r0=0x00825058 [0x00825059]:=>78000001 C
r2, r10, [r0, #4] ; 0x04 r2=0x005EEA38 r10=0x00000001 r0=0x00825058 [0x0082505C]:=>005eea38 C
r0, [r1, #0x58] r0=0x00825058 r1=0x00825014 [0x0082506C]:=>00825058 C

r0, r1, [r7, #0] ; 00 r0:0x00824F18=>0x00800C30 r1:0x00825014=>0x005EEA30 r7=0x0081FC74 [0x0081FC74]:=>00800c30 C
r1, [r0, #0x4] r1=0x005EEA30 r0=0x00800C30 [0x00800C34]:=>005eea30 C
r1, [r7, #0x8] r1:0x005EEA30=>0x00000008 r7=0x0081FC74 [0x0081FC7C]:=>00000008 C

r1, [r4, #0] r1:0x00000000=>0x005EEA30 r4=0x0081FC90 [0x0081FC90]:=>005eea30 C
r2, r1, #7 ; 0x00000007 r2:0x003D1CBE=>0x00000000 r1=0x005EEA30 C

r1, r0, [r13, #4] ; 0x04 r1=0x005EEA30 r0=0x000000FF r13=0x79AF510D [0x79AF5111]:=>unknown C
r0, [SP, #0x24] r0:0x000000FF=>0x00000001 C
r1, r1, #0x03 r1:0x005EEA30=>0x000BDD46 cpsr:0x20000133=>0x00000133 C

r0, r7, #0x01 r0:0x00000006=>0x0005EEA3 r7=0x000BDD46 cpsr:0x20000133=>0x00000133 C
r11, [r4, #26] ; 0x01a r11=0x00000000 r4=0x00824F3C [0x00824F56]:=>00010000 C
r8, #0 ; 00000000 r8=0x00000001 cpsr:0x00000133=>0x20000133 C
r0, [r4, #0x1c] r0=0x0005EEA3 r4=0x00824F3C [0x00824F58]:=>0005eea3 C
r1, r7, #1 ; 0x00000001 r1:0x000000A4=>0x00000000 r7=0x000BDD46 C
r5, [r4, #0x20] r5=0x000000FF r4=0x00824F3C [0x00824F5C]:=>000000ff C
r3, r4, #20 ; 0x00000014 r3:0x00000000=>0x00824F50 r4=0x00824F3C C
r0, [SP, #0x2c] r0:0x0005EEA3=>0x00000000 C

r1!, {r3, r4, r12, r14} r1:0x00824F50=>0x00824F60 r3:0x00000021=>0x000000A4 r4:0x00000022=>0x00000106 r12:0x000000A8=>0x0005EEA3
r14=0x9997FC4B C
r2, r2, #0x20 r2:0x00000000=>0xFFFFFFE0 cpsr:0x60000113=>0x80000113 C
r0!, {r3, r4, r12, r14} r0:0x418004B0=>0x418004C0 r3=0x000000A4 r4=0x00000106 r12=0x0005EEA3 r14=0x9997FC4B C
r1!, {r3, r4, r12, r14} r1:0x00824F60=>0x00824F70 r3:0x000000A4=>0x000000A8 r4:0x00000106=>0x804C5200 r12:0x0005EEA3=>0x00000060
r14=0x9997FC4B C

r1, [r4, #0] r1:0x00824F70=>0x005EEA30 r4=0x0081FC90 [0x0081FC90]:=>005eea30 C
r0, r1, r8 r0:0x00000001=>0x005EEA38 r1=0x005EEA30 r8=0x00000008 C
r0, [r4, #0] r0=0x005EEA38 r4=0x0081FC90 [0x0081FC90]:=>005eea38 C
r1, [r4, #0x4] r1:0x005EEA30=>0x00000008 r4=0x0081FC90 [0x0081FC94]:=>00000008 C
r0, r1, r8 r0:0x005EEA38=>0x00000000 r1=0x00000008 r8=0x00000008 C

READrel_19076(char flash_channel0..7, MSB<<13, requestbase+8, FlashCMD|0x30, MSB<<8)

Reading from Flash:
0x20480000=Command|0x30
0x20480004=Flash-Address
0x20480014|=1
Sleep(100);
0x20480000|=0x100

Wichtig:
A4 = 0 .. 15