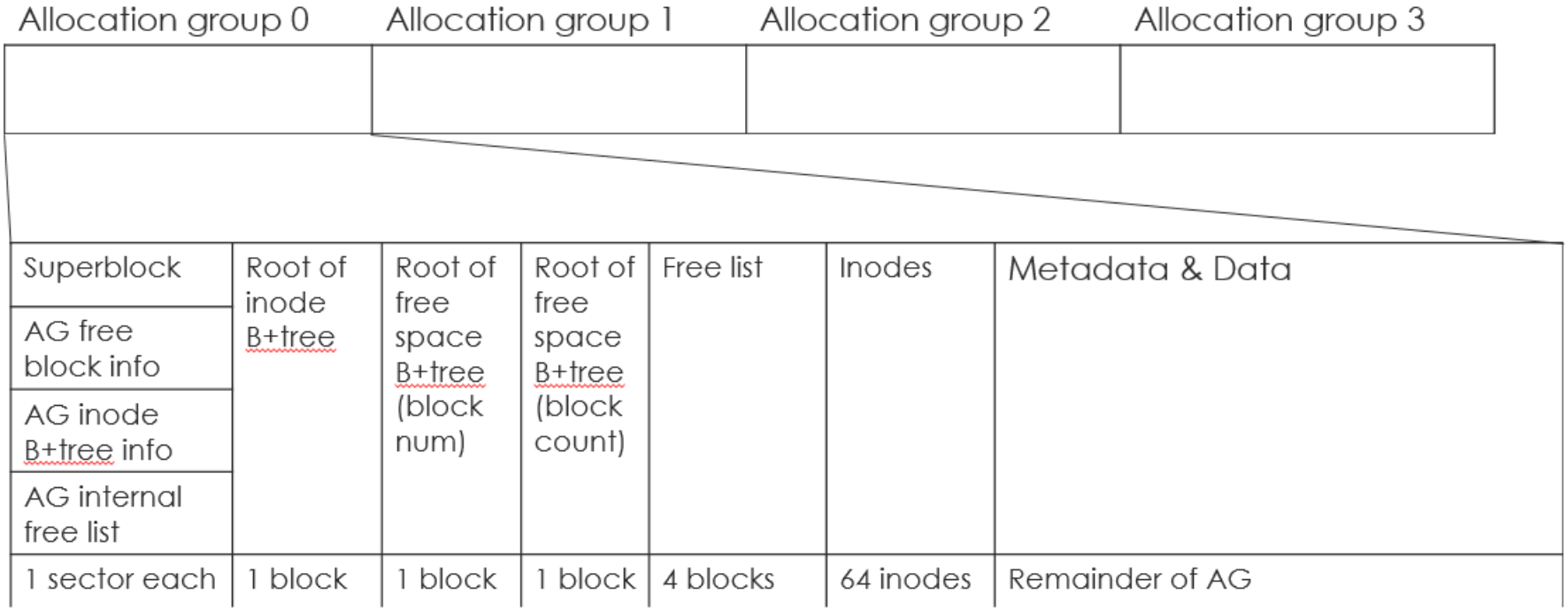
XFS Cheat sheet

References: http://ftp.ntu.edu.tw/linux/utils/fs/xfs/docs/xfs_filesystem_structure.pdf
https://righteousit.wordpress.com/2018/05/21/xfs-part-1-superblock/

Volume layout



Superblock

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	Signature XFSB				Block Size				Total blocks in filesystem							
0x10	Num blocks in real-time device								Num extents in real-time device							
0x20	UUID															
0x30	First block of journal								Root directory inode							
0x40	Real-time extents bitmap inode								Real-time bitmap summary inode							
0x50	Real-time extent size (blocks)				AG size (blocks)				Number of AGs				Num of real-time bitmap blocks			
0x60	Num of journal blocks				Version num		Sector size		Inode size		Inodes/ block		FS name			
0x70	FS name								s*	t*	u*	v*	w*	x*	y	z
0x80	Global count for num inodes								Global count for free inodes							
0x90	Global count for free data blocks								Global count for free real-time extents							
0xA0	Inode for user quotas								Inode for group or project quotas							
0xB0	Quota flags		RO flag	zero	Inode chunk alignment				Stripe/RAID unit in blocks				Stripe or RAID width in blocks			
0xC0	q	r	Log sector size[1]		The log device's stripe or raid unit size[1]				Additional version\features flags				duplicate of Additional version flags			
0xD0	RW compatible flags (currently unused)				RO compatible flags				RW required features				RW required log features			
0xE0	CRC				Sparse inode alignment				Project quote inode							
0xF0	Log sequence number of last SB update								Metadata UUID (if XFS_SB_FEAT_INCOMPAT_META_UUID set)							
0x100	Metadata UUID (cont…)								Root of real-time reverse mapping B+tree (if real-time enabled)							

***** these are all $\log_2$ size values i.e 2^{value}

s = $\log_2$ (Block size)
t = $\log_2$ (sector size)
u = $\log_2$ (inode size)
v = $\log_2$ (inodes per block)
q = $\log_2$ (directory block allocations in fsblocks)
w = $\log_2$ (AG size)
x = $\log_2$ (real-time extents)
y = flag filesytem is being created
z = Maximum % of filesystem space that can be used for inodes
r [1]= $\log_2$ (subvolume sector size)

[1] only applies if journaling log is on separate disk

Inode

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	magic num (IN)		File mode		Ver[2]	DF type	Unused		Owner UID				GID			
0x10	Number of links				Project ID				Padding						Increment on flush	
0x20	atime				atime nanoseconds				mtime				mtime nanoseconds			
0x30	ctime				ctime nanoseconds				file size							
0x40	Number blocks				Extent size				Number extents							
0x50	S		T	U	DMAPI event mask				DMAPI state		flags		Generation number			
0x60	next unlinked ptr[3]				CRC32				Count of changes to attributes							
0x70	log sequence num								Extended flags							
0x80	Write extent size hint								Reserved for future use							
0x90	btime				btime nanoseconds				inode number of this inode							
0xA0	UUID															

S = Number of extended attribute extents
T = Offset to extended attributes (multiply by 8)
U = Extended attribute fork format

Unlike EXT btime or creation time of the inode is recorded. Note that official documentation refers to this and change time as ctime.

File mode values

Value	Binary	hex
Other execute	0000000000000001	0x0001
Other write	0000000000000010	0x0002
Other read	0000000000000100	0x0004
Group execute	0000000000001000	0x0008
Group write	0000000000010000	0x0010
Group read	0000000000100000	0x0020
Owner execute	0000000001000000	0x0040
Owner write	0000000010000000	0x0080
Owner Read	0000000100000000	0x0100
Sticky bit	0000001000000000	0x0200
Set process GID	0000010000000000	0x0400
Set process UID	0000100000000000	0x0800

[2] Version refers to the inode version, this is different to the XFS version. At the time of writing Inode V3 is used in XFSv5.
[3] Used if the file has been marked for deletion but is still in use. The filesystem maintains a linked list of files in this state. In the event of a crash they will be deleted.

Value	Binary	hex
FIFO	0001000000000000	0x1000
Character device	0010000000000000	0x2000
Directory	0100000000000000	0x4000
Block device	0110000000000000	0x6000
Regular file	1000000000000000	0x8000
Symbolic link	1010000000000000	0xA000
Socket	1100000000000000	0xC000

Short form Directory Header

Offset	Length	Value
0	1	Number of directory entries
1	1	Number of directory entries using 64bit inode
2	4	Inode of parent

Directory Entry

Offset	Length	Value
0	1	Length of filename
1	2	Entry offset in non short form directory
2	varies	Filename
2+FN len	1	File type
3+FN len	4 or 8	Absolute inode address