How MUX Mapping Works

Rob MacLeod, Bonnie Punske, Bruce Steadman

June 25, 2017

1 Introduction

This is an attempt to describe the layout of channels in the CVRTI multiplexers (MUX) as they come out on the oscilloscope displays and in the data stream of the acquisition files. The description is targeted at the large MUXs we now use routinely so is not likely to be correct for a small (*e.g.,* 256 channel) MUX system if you are stuck using one.

We begin with a short intro in Section 2 and then describe in Section 3 some simple tools for making your own MUX mapping files. For those who really want to know, you can find all the ugly details in Section 4.

2 The Rules of the Game

1. For the 512-channel setup, channels alternate between the two 256-channel banks: the odd numbered channels are from bank #1 and the even numbered from bank #2. As a result, the order of stored channels for bank #1 is 1, 3, 5, 7, . . . , 509, 511 and that for bank #2 is 2, 4, 6, 8,. . . , 510, 512.

2. In the 1024-channel setup, there are still banks of 256, and channels alternate between master and slave MUX's: master has odd numbers, slave as even. Within the MUX's, channel order also alternates so that the final pattern is:

   (a) MUX #1, bank 1
   (b) MUX #2, bank 1
   (c) MUX #1, bank 2
   (d) MUX #2, bank 2

   The resulting order of stored channels is:

   (a) MUX #1, bank 1: 1, 5, 9, . . . , 1017, 1021
   (b) MUX #1, bank 2: 3, 7, 11, . . . , 1019, 1023
   (c) MUX #2, bank 1: 2, 6, 10, . . . , 1018, 1022
   (d) MUX #2, bank 2: 4, 8, 12, . . . , 1020, 1024

Another way to picture the mapping between leads of each bank and the channels in the multiplexed datastream is by the following equations:

Bank	Mapping	Example
Master #1	$ch = (l-1)*4+1$	1, 5, 9, 13,. . . ,1017, 1021
Master #2	$ch = (l-1)*4+3$	3, 7, 11, 15,. . . ,1023
Slave #1	$ch = (l-1)*4+2$	2, 6, 10, 14,. . . ,1022
Slave #1	$ch = (l-1)*4+4$	4, 8, 12, 16,. . . ,1024

Figure 1 illustrates this scheme in a diagram.

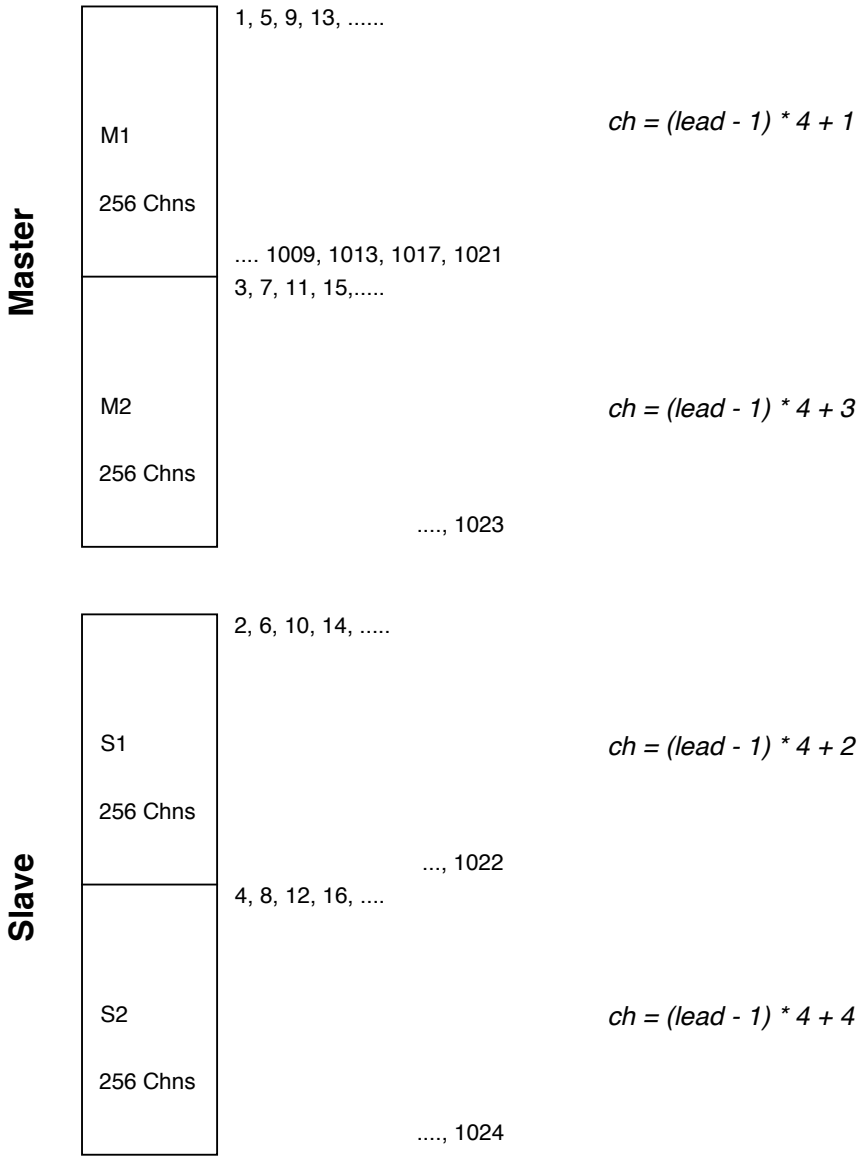

Figure 1: Mux mapping for the 1025 channel MUX.

3 Tools for Making MUX Mapping Files

We have some software for creating MUX mapping files that should take care of most needs. The program (script actually) that does this is called `makemuxmapping` and it has the following format:

```
Usage: makemuxmapping.sh -t numtank -s numsock -n numneedles -m 512/1024 -f
        -t to set number of tank electrodes
        -s to set number of sock electrodes
        -n to set number of needles
        -m to set MUX config (512 or 1024 leads)
        -f for a full mapping file (with padding)
```

-t numtank sets the number of tank electrodes. (*e.g.,* 192 or 374 for andy3).

-s numsock sets the number of sock electrodes (*e.g.,* 128, 490)

-n numneedles sets the number of needles, and we assume that each needle has 10 electrodes.

-m 512/1024 selects between a 512 or 1024 channel configuration of the MUX('s)

-f if present, this requests a full MUX mapping file, *i.e.,* one that contains padding to fill the entire 512 or 1024 channels. This is required for the data acquisition program.

3.1 Filename conventions

The filename conventions that makemuxmapping assumes are:

1. File extension is `.mux` (I decided this because the filenames were otherwise getting too long for the Mac)
2. First part of file name suggests which surface comes first in the mapping, *e.g.,* andy3, sock, needles.
3. What follows are numbers and letters to indicate the number and type of leads, *e.g.,* 490s = 490-lead sock, 10n = 10 needles, 374t = 374 tank electrodes.
4. The underscore _ joins the segments of the filenames.

See the next section for examples of filenames.

3.2 Examples

To make an andy3 mapping with 374 leads and a 490 lead sock in a 1024 MUX.

```
> makemuxmapping.sh -t 374 -s 490 -m 1024

Wrote 192 channels of tank
Wrote 182 channels of secondary tank
Wrote 490 channels of sock
For a total of 864 channels
Finished with andy3_374t_490s_1024.mux
```

To have the same contents, but padded for use in the experiment:

```
> makemuxmapping.sh -t 374 -s 490 -m 1024 -f

Wrote 192 channels of tank
Wrote 182 channels of secondary tank
Wrote 490 channels of sock
Wrote 160 channels of end fill
For a total of 1024 channels
Finished with andy3_374t_490s_1024_full.mux
```

To make a mapping file for a 490 lead sock in the 512 channel configuration:

```
> makemuxmapping.sh -s 490 -m 512 -f

Wrote 490 channels of sock
Wrote 22 channels of end fill
For a total of 512 channels
Finished with sock_490s_full.mux
```

And to now add 22 needles to this configuration

```
> makemuxmapping.sh -s 490 -n 22 -m 512 -f

Wrote 490 channels of sock
Wrote 22 channels of end fill
For a total of 512 channels
Finished with sock_490s_22n_full.mux
```

4 Untangling the Mess: The Ugly Details

In order to see the data we record in the proper order, or even to monitor leads during an experiment, we need to have some methods for untangling all these leads. For that we have "mux mapping" files, and there are a lot of them, one for each type of electrode configuration.

Mux mapping files contain a list of channel numbers. The **order** of the channels is the same as the order of the channels we want to have (*e.g.,* in an output file). The **value** of the channels in the list indicate the source of that channel in the MUX itself.

Perhaps the best way to see this in action is in an example. Here is a mux mapping file for a 128-lead sock and 22 10-pole needles. We break the mux mapping file into sections to make this ordering clear.

First the 128 sock leads:

```
512 channels
  1   3   5   7   9  11  13  15
 17  19  21  23  25  27  29  31
 33  35  37  39  41  43  45  47
 49  51  53  55  57  59  61  63
 65  67  69  71  73  75  77  79
 81  83  85  87  89  91  93  95
 97  99 101 103 105 107 109 111
113 115 117 119 121 123 125 127
129 131 133 135 137 139 141 143
145 147 149 151 153 155 157 159
161 163 165 167 169 171 173 175
177 179 181 183 185 187 189 191
193 195 197 199 201 203 205 207
209 211 213 215 217 219 221 223
225 227 229 231 233 235 237 239
241 243 245 247 249 251 253 255
```

Followed by the 22, 10-pole needles, which require 220 channels. These are plugged into the second bank:

```
  2   4   6   8  10  12  14  16
 18  20  22  24  26  28  30  32
 34  36  38  40  42  44  46  48
 50  52  54  56  58  60  62  64
 66  68  70  72  74  76  78  80
 82  84  86  88  90  92  94  96
 98 100 102 104 106 108 110 112
114 116 118 120 122 124 126 128
130 132 134 136 138 140 142 144
146 148 150 152 154 156 158 160
162 164 166 168 170 172 174 176
178 180 182 184 186 188 190 192
194 196 198 200 202 204 206 208
210 212 214 216 218 220 222 224
226 228 230 232 234 236 238 240
242 244 246 248 250 252 254 256
258 260 262 264 266 268 270 272
274 276 278 280 282 284 286 288
290 292 294 296 298 300 302 304
306 308 310 312 314 316 318 320
```

```
322  324  326  328  330  332  334  336
338  340  342  344  346  348  350  352
354  356  358  360  362  364  366  368
370  372  374  376  378  380  382  384
386  388  390  392  394  396  398  400
402  404  406  408  410  412  414  416
418  420  422  424  426  428  430  432
434  436  438  440
```

Now, we fill out the rest of the mapping file with all the other leads, starting with the remaining ones from bank #1.

```
                      257  259  261  263
265  267  269  271  273  275  277  279
281  283  285  287  289  291  293  295
297  299  301  303  305  307  309  311
313  315  317  319  321  323  325  327
329  331  333  335  337  339  341  343
345  347  349  351  353  355  357  359
361  363  365  367  369  371  373  375
377  379  381  383  385  387  389  391
393  395  397  399  401  403  405  407
409  411  413  415  417  419  421  423
425  427  429  431  433  435  437  439
441  443  445  447  449  451  453  455
457  459  461  463  465  467  469  471
473  475  477  479  481  483  485  487
489  491  493  495  497  499  501  503
505  507  509  511
```

and then all those left from bank #1.

```
                      442  444  446  448
450  452  454  456  458  460  462  464
466  468  470  472  474  476  478  480
482  484  486  488  490  492  494  496
498  500  502  504  506  508  510  512
```

The reason for padding the file this way is that the data acquisition program requires the number of entries to match the number of channels—this way there are no unidentified channels. For remapping the data to create time series files, it makes more sense to shed all the empty channels and so the resulting mux mapping file contains only those necessary (in this case 128 + 220 = 348. The file looks like this:

```
348 channels
```

1	3	5	7	9	11	13	15
17	19	21	23	25	27	29	31
33	35	37	39	41	43	45	47
49	51	53	55	57	59	61	63
65	67	69	71	73	75	77	79
81	83	85	87	89	91	93	95
97	99	101	103	105	107	109	111
113	115	117	119	121	123	125	127
129	131	133	135	137	139	141	143
145	147	149	151	153	155	157	159
161	163	165	167	169	171	173	175
177	179	181	183	185	187	189	191
193	195	197	199	201	203	205	207
209	211	213	215	217	219	221	223
225	227	229	231	233	235	237	239
241	243	245	247	249	251	253	255
2	4	6	8	10	12	14	16
18	20	22	24	26	28	30	32
34	36	38	40	42	44	46	48
50	52	54	56	58	60	62	64
66	68	70	72	74	76	78	80
82	84	86	88	90	92	94	96
98	100	102	104	106	108	110	112
114	116	118	120	122	124	126	128
130	132	134	136	138	140	142	144
146	148	150	152	154	156	158	160
162	164	166	168	170	172	174	176
178	180	182	184	186	188	190	192
194	196	198	200	202	204	206	208
210	212	214	216	218	220	222	224
226	228	230	232	234	236	238	240
242	244	246	248	250	252	254	256
258	260	262	264	266	268	270	272
274	276	278	280	282	284	286	288
290	292	294	296	298	300	302	304
306	308	310	312	314	316	318	320
322	324	326	328	330	332	334	336
338	340	342	344	346	348	350	352
354	356	358	360	362	364	366	368
370	372	374	376	378	380	382	384
386	388	390	392	394	396	398	400
402	404	406	408	410	412	414	416
418	420	422	424	426	428	430	432
434	436	438	440				

For the 1024 channel configuration, the file for the same leads looks **completely** different! If we left all the connectors in the same place and just attached the second MUX, the resulting mux mapping file would be (the version without padding) as follows:

```
384 channels
  1   5   9  13  17  21  25  29
 33  37  41  45  49  53  57  61
 65  69  73  77  81  85  89  93
 97 101 105 109 113 117 121 125
129 133 137 141 145 149 153 157
161 165 169 173 177 181 185 189
193 197 201 205 209 213 217 221
225 229 233 237 241 245 249 253
257 261 265 269 273 277 281 285
289 293 297 301 305 309 313 317
321 325 329 333 337 341 345 349
353 357 361 365 369 373 377 381
385 389 393 397 401 405 409 413
417 421 425 429 433 437 441 445
449 453 457 461 465 469 473 477
481 485 489 493 497 501 505 509
  2   6  10  14  18  22  26  30
 34  38  42  46  50  54  58  62
 66  70  74  78  82  86  90  94
 98 102 106 110 114 118 122 126
130 134 138 142 146 150 154 158
162 166 170 174 178 182 186 190
194 198 202 206 210 214 218 222
226 230 234 238 242 246 250 254
258 262 266 270 274 278 282 286
290 294 298 302 306 310 314 318
322 326 330 334 338 342 346 350
354 358 362 366 370 374 378 382
386 390 394 398 402 406 410 414
418 422 426 430 434 438 442 446
450 454 458 462 466 470 474 478
482 486 490 494 498 502 506 510
514 518 522 526 530 534 538 542
546 550 554 558 562 566 570 574
578 582 586 590 594 598 602 606
610 614 618 622 626 630 634 638
642 646 650 654 658 662 666 670
674 678 682 686 690 694 698 702
706 710 714 718 722 726 730 734
738 742 746 750 754 758 762 766
```

```
770  774  778  782  786  790  794  798
802  806  810  814  818  822  826  830
834  838  842  846  850  854  858  862
866  870  874  878
```