Getting the groups:

From Program: ('audio', 'membus', (5, 36, 39, 41, 48, 49, 58, 59))

Manual check: 5 39 49 59 48 58 36 41

Match ✅

('cache0', 'cache1', (7, 8, 11, 12, 16, 17, 19, 20)) 8 11 17 20 7 12 16 19

('cache0', 'cpu0', (0, 2, 25, 26)) 25 26 0 2

('cache0', 'membus', (9, 10, 13, 18, 21, 24, 27, 30)) 13 18 21 27 9 10 24 30

('cache1', 'cpu1', (1, 3, 29, 32)) 29 32 1 3

('cache1', 'membus', (14, 22, 28, 31)) 14 22 28 31

('gfx', 'membus', (4, 35, 38, 40, 54, 55, 56, 57)) 4 38 55 57 54 56 35 40

('mem', 'membus', (15, 23, 33, 34)) 23 34 15 33

('membus', 'uart', (42, 43, 46, 47, 50, 51)) 46 50 43 42 47 51

('membus', 'usb', (6, 37, 44, 45, 52, 53)) 44 52 37 6 45 53

Testing

Example Group: ('cache0', 'cache1', (7, 8, 11, 12, 16, 17, 19, 20)) 8 11 17 20 7 12 16 19

Sequence:

[16, 20, 8, 12, 17, 19, 8, 12, 8, 12, 16, 7, 20, 11, 17, 19, 7, 7, 11, 11, 16, 17, 20, 19, 17, 19, 7, 11, 17, 19, 7, 11, 8, 12, 17, 19, 17, 19, 17, 8, 12, 19, 8, 12, 16, 20, 17, 19, 17, 19, 8, 12, 17, 19, 7, 11, 17, 19, 7, 11, 7, 11, 8, 12, 8, 12, 17, 19, 8, 12, 16, 20, 8, 12, 16, 20, 17, 19, 16, 20, 16, 20, 16, 20, 7, 11, 17, 8, 12, 19, 17, 19, 8, 12, 16, 20, 17, 17, 19, 19, 7, 11, 7, 11, 8, 12, 7, 11, 8, 16, 12, 20, 8, 12, 7, 11, 8, 12, 16, 20, 17, 19, 8, 12, 8, 12, 7, 11, 7, 11, 17, 19, 16, 20, 17, 19, 16, 8, 12, 20, 7, 11, 16, 20, 7, 11, 16, 20, 16, 20, 8, 12, 16, 20, 8, 12, 7, 7, 11, 11, 8, 12, 8, 12, 7, 11, 7, 11, 16, 20, 7, 11, 7, 11, 16, 20, 16, 20, 8, 12, 17, 19, 17, 8, 12, 19, 7, 11, 16, 20, 7, 11, 7, 11, 8, 12, 17, 19, 17, 19, 8, 12, 7, 11, 17, 19, 17, 19, 7, 8, 11, 12, 16, 20, 7, 11, 16, 20, 16, 20, 17, 19, 8, 12, 16, 20, 7, 11, 16, 20, 16, 20, 16, 20, 16, 20, 17, 19, 17, 19]

- Yes, the sequence only includes numbers from the group ✅

TRACE LIST trace-small-5.txt

[3, 16, 20, 22, 52, 53, 31, 32, 0, 8, 12, 13, 54, 55, 24, 25, 2, 0, 17, 19, 8, 12, 13, 15, 21, 52, 23, 53, 30, 24, 26, 25, 1, 0, 8, …etc]

- And it follows the order in the trace ✅

Extracted sequences for trace-small-5.txt

'audio', 'membus' [48, 49, 48, 49, 48, 49, 48, 49, 48, 49, 48, 49, 48, 49, 48, 49, 48, 49, 48, 49, 48, 49, 48, 49, 48, 49, 48, 49, 48, 49, 48, 49, 48, 48, 49, 49, 48, 49, 48, 49]

'cache0', 'cache1' [16, 20, 8, 12, 17, 19, 8, 12, 8, 12, 16, 7, 20, 11, 17, 19, 7, 7, 11, 11, 16, 17, 20, 19, 17, 19, 7, 11, 17, 19, 7, 11, 8, 12, 17, 19, 17, 19, 17, 8, 12, 19, 8, 12, 16, 20, 17, 19, 17, 19, 8, 12, 17, 19, 7, 11, 17, 19, 7, 11, 7, 11, 8, 12, 8, 12, 17, 19, 8, 12, 16, 20, 8, 12, 16, 20, 17, 19, 16, 20, 16, 20, 16, 20, 7, 11, 17, 8, 12, 19, 17, 19, 8, 12, 16, 20, 17, 17, 19, 19, 7, 11, 7, 11, 8, 12, 7, 11, 8, 16, 12, 20, 8, 12, 7, 11, 8, 12, 16, 20, 17, 19, 8, 12, 8, 12, 7, 11, 7, 11, 17, 19, 16, 20, 17, 19, 16, 8, 12, 20, 7, 11, 16, 20, 7, 11, 16, 20, 16, 20, 8, 12, 16, 20, 8, 12, 7, 7, 11, 11, 8, 12, 8, 12, 7, 11, 7, 11, 16, 20, 7, 11, 7, 11, 16, 20, 16, 20, 8, 12, 17, 19, 17, 8, 12, 19, 7, 11, 16, 20, 7, 11, 7, 11, 8, 12, 17, 19, 17, 19, 8, 12, 7, 11, 17, 19, 17, 19, 7, 8, 11, 12, 16, 20, 7, 11, 16, 20, 16, 20, 17, 19, 8, 12, 16, 20, 7, 11, 16, 20, 16, 20, 16, 20, 16, 20, 17, 19, 17, 19]

'cache0', 'cpu0' [0, 25, 2, 0, 26, 25, 0, 25, 2, 26, 2, 26, 2, 26, 2, 26, 0, 2, 25, 26, 2, 26, 2, 0, 26, 25, 0, 25, 2, 26, 2, 26, 0, 25, 2, 26, 2, 26, 0, 25, 0, 25, 2, 26, 0, 25, 0, 25, 2, 26, 2, 0, 25, 26, 2, 26, 2, 26, 0, 25, 2, 2, 26, 26, 0, 25, 0, 25, 0, 25, 0, 25, 0, 25, 0, 25, 2, 0, 26, 25, 0, 25, 2, 26, 2, 26, 0, 25, 0, 25, 2, 26, 0, 25, 0, 25, 0, 25, 0, 25, 0, 25, 0, 25, 2, 2, 26, 26, 0, 2, 25, 26, 0, 25, 0, 25, 2, 2, 26, 26, 0, 2, 26, 25, 2, 26, 2, 26, 0, 25, 2, 26, 2, 26, 0, 25, 2, 26, 2, 26]

'cache0', 'membus' [13, 24, 13, 21, 30, 24, 13, 24, 21, 30, 21, 30, 21, 30, 13, 24, 21, 30, 21, 30, 13, 21, 30, 24, 21, 30, 21, 30, 21, 30, 21, 30, 13, 24, 13, 24, 21, 30, 13, 24, 21, 30, 13, 21, 24, 30, 21, 30, 21, 30, 13, 24, 13, 24, 13, 24, 21, 13, 30, 24, 13, 24, 21, 30, 21, 30, 13, 24, 13, 24, 13, 24, 13, 24, 13, 24, 13, 24, 21, 30, 13, 24, 13, 24, 21, 30, 21, 13, 30, 24, 21, 30, 13, 24, 13, 24, 21, 30, 21, 30]

'cache1', 'cpu1' [3, 32, 1, 29, 3, 1, 29, 32, 1, 1, 29, 29, 3, 32, 1, 29, 1, 29, 3, 32, 1, 29, 1, 29, 1, 29, 3, 32, 3, 32, 3, 32, 3, 32, 3, 1, 32, 29, 3, 32, 1, 29, 1, 29, 1, 29, 3, 32, 1, 29, 3, 32, 3, 32, 1, 29, 3, 32, 1, 29, 3, 32, 3, 32, 1, 29, 3, 32, 3, 32, 1, 29, 3, 32, 3, 32, 3, 32, 1, 1, 29, 29, 1, 29, 1, 29, 3, 1, 32, 29, 1, 29, 3, 32, 3, 32, 1, 3, 32, 29, 3, 32, 1, 29, 1, 29, 3, 32, 1, 29, 1, 29, 3, 32, 1, 29, 3, 32, 3, 32, 3, 32, 1, 3, 29, 32, 3, 32, 1, 3, 29, 32, 3, 32, 1, 29, 1, 29, 1, 29]

'cache1', 'membus' [22, 31, 22, 31, 14, 28, 14, 28, 14, 28, 14, 28, 14, 28, 14, 28, 22, 31, 22, 31, 22, 31, 22, 31, 14, 28, 22, 31, 14, 28, 14, 28, 14, 28, 22, 31, 14, 28, 22, 31, 14, 28, 14, 28, 22, 31, 14, 28, 22, 31, 14, 28, 22, 31, 22, 31, 22, 31, 14, 14, 28, 28, 14, 28, 14, 28, 14, 28, 22, 31, 22, 31, 14, 28, 22, 31, 14, 28, 14, 28, 14, 28, 14, 28, 22, 31, 22, 31, 22, 31, 22, 31, 22, 31, 22, 31, 22, 31, 22, 31]

'gfx', 'membus' [54, 55, 54, 55, 54, 55, 54, 55, 54, 55, 54, 54, 55, 55, 54, 55, 54, 55, 54, 55, 54, 55, 54, 55, 54, 55, 54, 55, 54, 55, 54, 55, 54, 55, 54, 55, 54, 55, 54, 55, 54, 55]

'mem', 'membus' [15, 23, 15, 23, 15, 23, 15, 23, 15, 15, 23, 23, 15, 23, 15, 23, 15, 23, 15, 23, 15, 23, 15, 23, 15, 23, 15, 23, 15, 23, 15, 23, 15, 23, 15, 23, 15, 23, 15, 23, 15, 23, 15, 23, 15, 23, 15, 23, 15, 23]

('membus', 'uart' [46, 47, 46, 47]

'membus', 'usb' [52, 53, 52, 53]