

## FC-AL-2 ARB Detection

FC-AL specifies the bit pattern transmitted for each primitive, but says nothing explicit about what received bit patterns are detected as specific primitives. While some have interpreted this as an implication that all 32 or 40 bits (4 bytes or characters) must be compared to detect a primitive, that implication has been missed by most readers and has not been implemented by existing FC-AL devices. Specific rules are being added to FC-AL-2 to address this confusion and/or problem.

The detection rules for most primitives have been non-controversial. For example, primitives that contain no parameter fields (e.g. RRdy, CLS) require checking all four characters or bytes (a so-called 40-bit comparison). Primitives such as OPN require checking three characters or bytes, the first two and the relevant address byte.

However the working group has been unable to reach consensus on the ARB primitive. We request that the plenary vote on the following to direct the document editor on what to include in FC-AL-2.

Note: Existing FC-AL devices from different vendors implement the following in conflicting ways. There is no set of rules consistent with all existing implementations (other than the current lack of rules). Reserved or invalid ARBs have been observed in loop testing, it is believed due to faulty transceivers with high error rates.

In the working group discussions, the consensus (compromise agreement?) on all of these detection rules is that they would be documented in FC-AL-2 as recommendations (“should”), with the intent (non-binding) that they become requirements in FC-AL-3. In each of the following I have asked for a separate motion on whether the detection rules should be a recommendation or requirement to obtain a recorded vote on this.

## 2. ARB(F0) detection (K28.5, D20.4, D16.7, D16.7)

ARB(F0) is used by the fairness algorithm. It is detected and responded to as a special case by several LPSM (Loop State Machine) transitions.

In the working group discussion yesterday, there was apparent consensus (no voiced objections) to recommending that all four characters or bytes must match exactly (a so-called 40-bit comparison) for detecting ARB(F0). The apparent consensus would imply voting Yes on motion 1a and No on motion 1b.

Motion 1a: Move that FC-AL-2 will either recommend (“should”) or require (“shall”) checking all four bytes or characters for an exact match to detect ARB(F0).

Vote Yes if you feel FC-AL-2 should give guidance on detecting ARB(F0).

Vote No if you feel FC-AL-2 should remain silent on this (as does FC-AL).

Motion 1b: Move that FC-AL-2 will require (rather than simply recommend) checking all four bytes or characters for an exact match to detect ARB(F0).

Vote Yes if you feel this shall be required to claim FC-AL-2 compliance.

Vote No if you feel that “old” FC-AL devices should be allowed to claim FC-AL-2 compliance, while giving guidance for new implementations.

## 3. Own ARB detection (K28.5, D20.4, AL\_PA, AL\_PA)

FC-AL ports detect their own ARB to win arbitration, plus a few special cases within the LPSM.

In both yesterday’s and the previous working group discussion, there was apparent consensus (no voiced objections) to recommending that all four characters or bytes must match exactly (a so-called 40-bit comparison) for detecting the port’s own ARB. The apparent consensus would imply voting Yes on motion 2a and No on motion 2b.

Motion 2a: Move that FC-AL-2 will either recommend (“should”) or require (“shall”) checking all four bytes or characters for an exact match for a port to detect its own ARB.

Vote Yes if you feel FC-AL-2 should give guidance on detecting a port’s ARB.

Vote No if you feel FC-AL-2 should remain silent on this (as does FC-AL).

Motion 2b: Move that FC-AL-2 will require (rather than simply recommend) checking all four bytes or characters for an exact match for a port to detect its own ARB.

Vote Yes if you feel this shall be required to claim FC-AL-2 compliance.

Vote No if you feel that “old” FC-AL devices should be allowed to claim FC-AL-2 compliance, while giving guidance for new implementations.

## 5. Invalid ARB Processing

Later motions address the rules for detecting valid ARB primitives. These motions address the handling of any ordered set that begins with the characters K28.5, D20.4 and is not a valid ARB. I am calling these “invalid ARB primitives” here, the eventual term used in FC-AL-2 may be different.

There are two different behaviors that need to be addressed, since most implementations perform them with different logic. One is whether an invalid ARB may be deleted as a fill word for clock skew management. This typically takes place before the port’s elasticity buffer in logic operating on the recovered receive clock. The other is LPSM operation, which typically takes place after the elasticity buffer in logic operating on the port’s local clock.

In the working group discussion yesterday, there was apparent consensus (no voiced objections) to recommending that the LPSM should replace invalid ARB primitives with the CFW (Current Fill Word, typically the most recent valid ARB or Idle). There was apparent consensus that this implied that invalid ARB primitives should be considered deletable fill words, since the CFW being substituted could have been deleted. The apparent consensus would imply voting Yes on motions 3a and 3c and No on motions 3b and 3d. At present FC-AL appears to imply that invalid ARB primitives are “other ordered sets” that are passed through unaltered by the LPSM. That interpretation of FC-AL would likely be more emphatic if we otherwise clarify the rules for ARB detection.

Motion 3a: Move that FC-AL-2 will either recommend (“should”) or require (“shall”) that the LPSM substitute the CFW for invalid ARB primitives.

Vote Yes if you feel FC-AL-2 devices should not propagate invalid ARB primitives.

Vote No if you feel FC-AL-2 devices should forward invalid ARB primitives unaltered.

Motion 3b: Move that FC-AL-2 will require (rather than simply recommend) that the LPSM substitute the CFW for invalid ARB primitives.

Vote Yes if you feel this shall be required to claim FC-AL-2 compliance.

Vote No if you feel that “old” FC-AL devices should be allowed to claim FC-AL-2 compliance, while giving guidance for new implementations.

Motion 3c: Move that FC-AL-2 will either recommend (“should”) or require (“shall”) that ports consider all ordered sets beginning with K28.5, D20.4 (a so-called “20-bit comparison”, including both valid and invalid ARB primitives) as fill words that may be deleted for clock skew management.

Motion 3d: Move that FC-AL-2 will require (rather than simply recommend) that ports consider all ordered sets beginning with K28.5, D20.4 as fill words that may be deleted for clock skew management.

## 7. Valid ARB Detection

Ports detect valid ARB primitives for implementing arbitration and updating the CFW (Current Fill Word). Arbitration operates by comparison of the loop address in the received ARB against the port's own loop address. This comparison is considered valid and acted upon if and only if the received ARB is valid. One significant implication of this is that only valid ARB primitives may become the CFW (Current Fill Word). A single received ARB primitive, if it becomes the CFW, may be replicated and transmitted numerous times before it is replaced by a later fill word.

In the working group discussion yesterday, there was apparent consensus that an ARB primitive be detected as valid if its last two characters or bytes (character or byte 3 and 4) are equal. However subsequent discussion revealed that there were differing interpretations on how "equality" was determined. An early vote was taken before this misunderstanding was recognized; its results seem irrelevant.

Once this issue was understood, the following options were identified:

- 1) An ARB is valid if (and only if) the decoded (8-bit) values of bytes 3 and 4 are identical. This is most easily implemented by an equality comparison after the 8b/10b decoder.
- 2) An ARB is valid if (and only if) the decoded (8-bit) values of bytes 3 and 4 are identical, and the value is one of the neutral disparity code bytes. There are 134 neutral disparity code byte values, listed in FC-AL Table 1, which include AL\_PA loop addresses plus the reserved and special AL\_PA values. This is most easily implemented by an equality comparison of the encoded 10-bit characters before the 8b/10b decoder.
- 3) Allowing either 1 or 2 above as an implementation option (vendor unique behavior).

It should be noted that while choice 1 above is most easily implemented by a comparison in the 8b domain and choice 2 is most easily implemented by a comparison in the 10b domain, either can be implemented in the other domain with modest amounts of additional logic (n.b., there was considerable controversy over what constituted a "modest amount", I state that as my own opinion in the belief that there is no adjective that would be agreed to).

The principal arguments made in support of the choices were:

1. Advocates of both 1 and 2 felt that their preference was simpler in their respective implementations.
2. The majority of existing designs that implement such checking follow choice 1.
3. Choice 2 is simpler for FL\_ports that do not otherwise have an 8b/10b decoder.

Three votes were taken, all with one vote per company. The first was taken shortly before lunch, at which time at least one voter complained that he needed to check with his office before voting. Regardless, the vote was taken with results 10 for choice 1, 8 for choice 2 and 3 for choice 3 (each company allowed to vote for one choice).

The other two votes were taken after lunch, approximately 20 to 30 minutes after the stated time for reconvening. The second vote allowed for each company to vote for one of the three choices, results were 7 for choice 1, 4 for choice 2 and 3 for choice 3. The third vote allowed for only choices 1 and 2, results were 9 for choice 1 and 5 for choice 2.

Motion 4a: Move that FC-AL-2 either recommend (“should”) or require (“shall”) that an ARB primitive is detected as K28.5, D20.4 and identical values in bytes 3 and 4. All 256 data byte values in bytes 3 and 4 are permitted and should/shall be detected as valid provided that the two bytes are identical.

Motion 4b: Move that FC-AL-2 either recommend (“should”) or require (“shall”) that an ARB primitive is detected as K28.5, D20.4, identical values in bytes 3 and 4 where that value is one of the 134 neutral disparity byte values listed in [FC-AL] Table 1.

Motion 4c: Move that FC-AL-2 either recommend (“should”) or require (“shall”) that an ARB primitive is detected as K28.5, D20.4 and identical values in bytes 3 and 4. FC-AL-2 ports may optionally choose to require, for detection of ARB, that the value in bytes 3 and 4 be one of the 134 neutral disparity byte values listed in [FC-AL] Table 1.

Motion 4d: Whichever of 4a to 4c is approved (if any), FC-AL-2 will require (as opposed to simply recommend) the result.