Table 1: Phase Table Data Structure

m	$j = 0$	1	2	Name
0	-	-	-	p0
1	-	-	-	p1
2	-	+	-	p2
3	-	+	+	p3
4	+	-	-	p4
5	+	-	-	p5
6	+	+	-	p6
7	+	+	+	p7

1 Multiple Phase

The XTK is designed to handle multiple geometry descriptions with recursive cutting capabilities. This is an enabling technology to analyze complex multi-material models. The XTK assumes that the number of material phases is related to the number of geometry descriptions with the following relationship.

$$n_m = 2^{n_\phi} \tag{1}$$

where n_m represents the number of material phases and n_Φ represents the number of geometry descriptions.

2 Phase Table

In order to enable convenient lookup within the code, the phase index lookup table needs to be structured in a particular way which is demonstrated for 3 geometry descriptions in Table 1.

When the negative signs are interpreted as a 0 and the positive signs as 1 the location of a given phase can be computed as follows:

$$i = \sum_{j=0}^{n_\phi} \frac{2^{n_\phi}}{2^{j+1}} P_j \tag{2}$$

where P_j is the phase value on an entity at a given phase j. It is important to note 2^{n_ϕ} is constant and stored as a member variable in the phase table rather than an the fly computation . This lookup method provides fast access to the correct index without a search function. Once the XTK is passed the phase lookup data, it performs an order check and will throw an error if the data does not adhere to the structure found in Table 1. If providing the information in this order becomes a hassle, there could be a sorting algorithm to automatically do this with little consequence because it would be done once at the beginning of execution. In addition to allowing for user provided phase tables, the inverse of , i can be taken to automatically generate the table for a given n_ϕ and j.

$$P_{ij} = \tag{3}$$

In addition to the phase data, the phase table needs a phase name to associate with the phase index. Inside of the code, the phase index should exclusively be used to prevent unnecessary degradation of performance. The names

are used to export the conformal mesh to an external mesh library (i.e. STK). XTK provides some default options but allows for manual declaration and duplicate names (multiple void phase conditions).