

SUPPLEMENTARY MATERIALS

Structure solution and refinement of stacking faulted NiCl(OH)

Sebastian Bette^[a], Robert E. Dinnebier^[b] and Daniela Freyer^[a]

[a] TU Bergakademie Freiberg, Institute of Inorganic Chemistry, Leipziger Strasse 29, 09596 Freiberg, Germany

[b] Max Planck Institute for Solid State Research, Heisenbergstrasse 1, 70569 Stuttgart, Germany

Keyword: alkaline nickel(II)-chlorides, X-ray powder diffraction, stacking faults, microstructure of layered compounds, IR-spectroscopy

Abstract Two samples of pure NiCl(OH) were produced by hydrothermal synthesis and characterized by chemical analysis, IR-spectroscopy, high resolution laboratory X-ray powder diffraction and scanning electron microscopy. Layers, composed of edge sharing, distorted NiCl_{6x}(OH)_{6-6x} octahedra were identified as the main building blocks of the crystal structure. NiCl(OH) is isostructural to CoOOH and crystallizes in space group $R\bar{3}m$ ($a = 3.2606(1)$ Å, $c = 17.0062(9)$ Å). Each sample exhibits faults in the stacking pattern of the layers. Crystal intergrowth of (A γ B)(B α C)(C β A) and (A γ B)(A γ B) (C6-like, β -Ni(OH)₂ related) stacked layers was identified as the main feature of the microstructure of NiCl(OH) by *DIFFaX* simulations. A recursion routine for creating distinct stacking patterns of rigid body like layers in real space with distinct faults (global optimization) and a Rietveld compatible approach (local optimization), was realized and implemented into a *macro* for the program TOPAS, for the first time. This routine enables a recursive creation of supercell containing (A γ B)(B α C)(C β A), (A γ B)(A γ B) and (C β A)(B α C)(A γ B) stacking patterns, according to user defined transition probabilities. Hence it is an enhancement of the few previously published Rietveld compatible approaches. This routine was applied successfully to *create* and *adopt* a detailed microstructure model to the *measured* data of two stacking faulted NiCl(OH) samples.

The obtained microstructure models were supported by high resolution SEM images.

Appendix A. Additional tables

Table S 1. Crystallographic and Rietveld refinement data for NiCl(OH) at ambient conditions.

Compound name	NiCl(OH)
Molecular formula	NiCl(OH)
Sum formula	NiOClH
Molecular weight (g/mol)	111.15
Space group	$R\bar{3}m$
Z	3
$a / \text{\AA}$	3.2606(1)
$c / \text{\AA}$	17.0062(9)
$V / \text{\AA}^3$	156.58(1)
$\rho_{\text{calc}} / \text{g} \cdot \text{cm}^{-3}$	3.66
Wavelength / \AA	0.70930
$R\text{-exp} / \%$ *	3.55
$R\text{-p} / \%$ *	5.03
$R\text{-wp} / \%$ *	6.70
$R\text{-}F^2 / \%$ *	2.00
gof*	1.89
Starting angle ($^\circ 2\theta$)	2.0
Final angle ($^\circ 2\theta$)	60.0
Step width ($^\circ 2\theta$)	0.005
Time/scan (hrs)	23
No. of variables	42

* $R\text{-exp}$, $R\text{-p}$, $R\text{-wp}$ and $R\text{-}F^2$ as defined in TOPAS (Bruker AXS) (Bruker, 2009)

Table S 2. Atomic coordinates and selected bond distances of NiCl(OH) at ambient conditions.

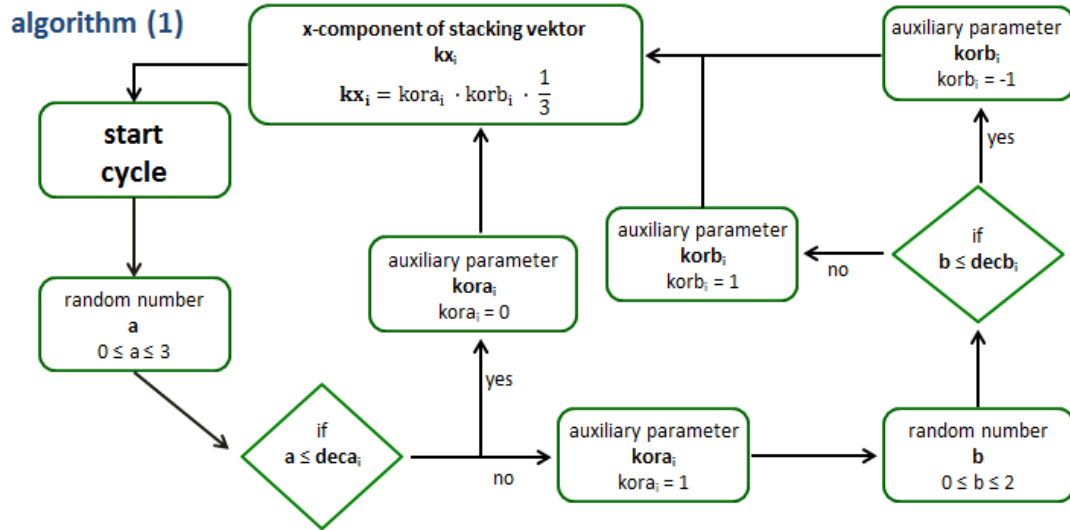
Atomic coordinats							
Atom	Wyck.	Site	S.O.F.	<i>x</i>	<i>y</i>	<i>z</i>	B /Å ²
Ni(1)	3 <i>a</i>	-3 <i>m</i>	1.0	0	0	0	1.21(2)
O(1)	6 <i>c</i>	3 <i>m</i>	0.5	0	0	0.3683(2)	0.44(3)
Cl(1)	6 <i>c</i>	3 <i>m</i>	0.5	0	0	0.4212(1)	0.44(3)
selected bond distances /Å							
Ni-O		1.974(1) (6x)					
Ni-Cl		2.404(1) (6x)					
Ni-Ni		3.261(1)					
O-O (min)		2.227(4)					
Cl-Cl (min)		3.261(1)					
O-Cl (min)		2.813(3)					

Appendix B. Detailed description of the recursion algorithm

In order to generate 0, $\frac{1}{3}$ and $-\frac{1}{3}$ as distinct values for kx_i incidentally (Figure S 1), the random number generator, implemented in TOPAS 4.2 was used to create a random number, **a**, for each layer, with the boundary condition: $0 \leq \mathbf{a} \leq 3$. A parameter, **deca_i**, is introduced for each layer, which decides whether an auxiliary parameter, **kora_i**, becomes 0 or 1 (Table S 3). Another random number **b** is created, having the boundary condition $0 \leq \mathbf{b} \leq 2$. The value of an auxiliary parameter, **korbi**, is set to 1, resp. -1 by decision (**decbi**). The x-component of the stacking vector **kx_i** is generated by multiplication of the auxiliary parameters (**kora_i**, **korbi**) with $\frac{1}{3}$. If **kora_i** is 0, then **kx_i** is 0, as well. If **kora_i** is 1, then it doesn't influence the product of **kora_i** and **korbi**, therefore and the value of **kx_i** is determined by **korbi** exclusively. A complete incidental generation of **kx_i** can be achieved by setting **deca_i** and **decbi** to 1, so the probabilities of **kx_i** = 0, $\frac{1}{3}$ and $-\frac{1}{3}$ are equal.

Table S 3. Overview of parameters used in algorithm (1)

parameter	description/ effect
\mathbf{kx}_i	x-component of the regular stacking vector \vec{k}_i , describing the atomic position of a layer i in the unit cell
\mathbf{a}	random number, $0 \leq \mathbf{a} \leq 3$, for generation of distinct values (0, $\frac{1}{3}$, $-\frac{1}{3}$) for \mathbf{kx}_i
\mathbf{b}	random number, $0 \leq \mathbf{b} \leq 2$, for generation of distinct values (0, $\frac{1}{3}$, $-\frac{1}{3}$) for \mathbf{kx}_i
\mathbf{deca}_i	decision parameter, constraint for setting \mathbf{kora}_i either to 1 or 0
\mathbf{decb}_i	decision parameter, constraint for setting \mathbf{korb}_i either to 1 or -1
\mathbf{kora}_i	auxiliary parameter, sets \mathbf{kx}_i to 0 (if $\mathbf{kora}_i = 0$)
\mathbf{korb}_i	auxiliary parameter, sets \mathbf{kx}_i to $\frac{1}{3}$ (if $\mathbf{kora}_i = 1$, $\mathbf{korb}_i = 1$) or $-\frac{1}{3}$ (if $\mathbf{kora}_i = 1$, $\mathbf{korb}_i = -1$)



probability at komplette coincidence: $\mathbf{deca}_i = \mathbf{decb}_i = 1$

$$p(\mathbf{kx}_i = 0) = \frac{\mathbf{deca}_i}{3} = \frac{1}{3}, \quad p\left(\mathbf{kx}_i = -\frac{1}{3}\right) = \frac{3-\mathbf{deca}_i}{3} \cdot \frac{\mathbf{decb}_i}{2} = \frac{3-1}{3} \cdot \frac{1}{2} = \frac{1}{3}, \quad p\left(\mathbf{kx}_i = \frac{1}{3}\right) = \frac{3-\mathbf{deca}_i}{3} \cdot \frac{2-\mathbf{decb}_i}{2} = \frac{3-1}{3} \cdot \frac{2-1}{2} = \frac{1}{3}$$

Figure S 1. Schematic representation of algorithm (1) for generation of distinct regular stacking vectors incidentally.

To obtain certain stacking patterns, like $(A\gamma B)(B\alpha C)(C\beta A)$, a definite sequence of stacking vectors has to be generated (Table 7). Therefore the parameters for decision (\mathbf{deca}_i and \mathbf{decb}_i) must be manipulated by the stacking vector of the preceding layer. This has to be done in a way in which the algorithm recognizes autonomously the category $((A\gamma B), (B\alpha C)$ or $(C\beta A))$ of the preceding layer and generates an adequate subsequent layer (e.g. $(C\beta A)$ after $(B\alpha C)$ for an $(A\gamma B)(B\alpha C)(C\beta A)$ stacking).

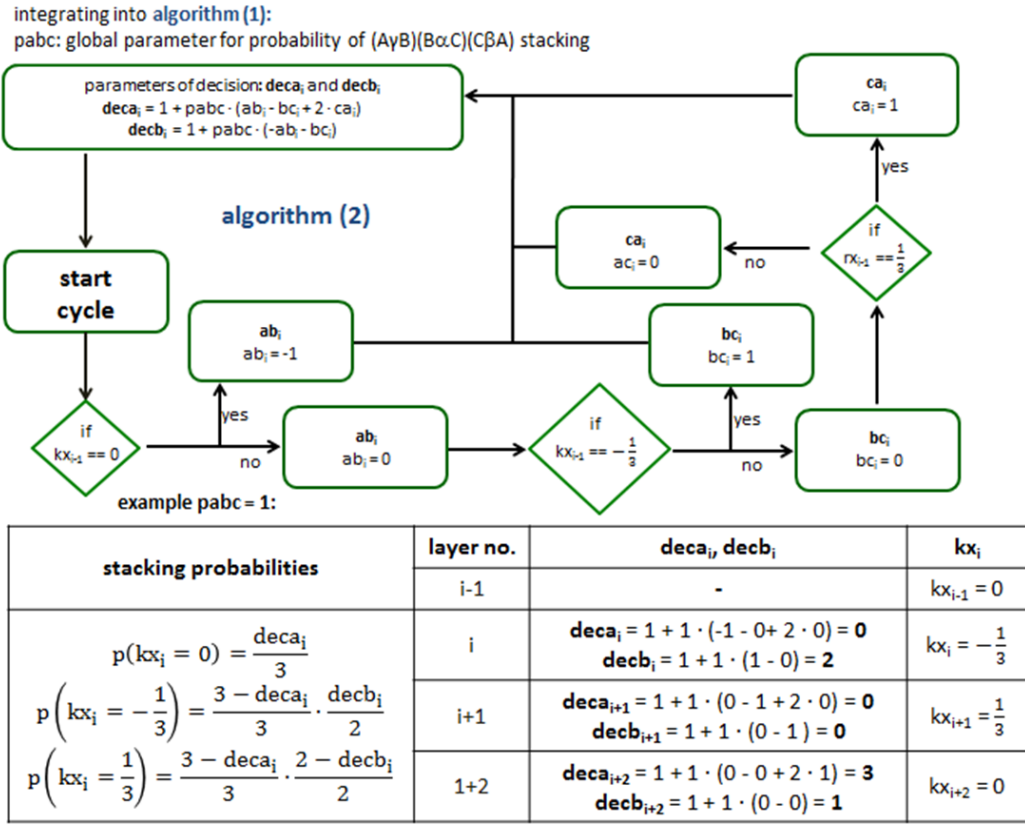


Figure S 2. Schematic representation of algorithm (2) for manipulation of the decision parameters $deca_i$ and $decb_i$, determined by the preceding layer, to generate a definite (A γ B)(B α C)(C β A) stacking pattern.

Table S 4. Overview of parameters used in algorithm (2).

parameter	description/ effect
ab_i	auxiliary parameter, sets deca_i to 0 and decb_i to 2 (if $kx_{i-1} = 0 \rightarrow$ preceding layer = (A γ B)-type) $\rightarrow kx_i = -\frac{1}{3}$, i.e. layer i = (B α C)-type
bc_i	auxiliary parameter, sets deca_i to 0 and decb_i to 0 (if $kx_{i-1} = -\frac{1}{3} \rightarrow$ preceding layer = (B α C)-type) $\rightarrow kx_i = \frac{1}{3}$, i.e. layer i = (C β A)-type
ca_i	auxiliary parameter, sets deca_i to 3 and decb_i to 1 (if $kx_{i-1} = \frac{1}{3} \rightarrow$ preceding layer = (C β A)-type) $\rightarrow kx_i = 0$, i.e. layer i = (A γ B)-type
pabc	parameter for probability of (A γ B)(B α C)(C β A) type stacking (pabc = 1 \rightarrow (A γ B)(B α C)(C β A) basis stacking)

Accordingly a second algorithm (2) is created and integrated into the first one (Figure S 2).

Therefore new auxiliary parameters (**ab_i**, **bc_i**, **ca_i**, Table S 4) are defined. The assignment of each parameter to the value 0 or 1 is determined by the stacking vector of the preceding layer

(\mathbf{kx}_{i-1}). As the auxiliary parameters are connected with \mathbf{deca}_i and \mathbf{decb}_i by addition, the value 0 represents the neutral element and the relating auxiliary parameter is therefore neglected. If one of these parameters is assigned to the value 1, it shifts the parameters of decision (\mathbf{deca}_i and \mathbf{decb}_i) in such a way, that a distinct stacking vector of the three basic vectors of an $(A\gamma B)(B\alpha C)(C\beta A)$ stacking ($= 0, \frac{1}{3}$ and $-\frac{1}{3}$) is generated with a probability of 1. Hence a definite stacking sequence $(0, -\frac{1}{3}, \frac{1}{3}, 0, \dots)$ is created. In addition a global parameter (\mathbf{pabc}) is defined. This parameter acts as a switch. If 1.0 is assigned to \mathbf{pabc} , an $(A\gamma B)(B\alpha C)(C\beta A)$ pattern is defined as a basic stacking pattern. Accordingly each shift in stacking pattern is defined as a fault. If a number, that is lower than 1.0 is assigned to \mathbf{pabc} , then this parameter defines the probability of an $(A\gamma B)(B\alpha C)(C\beta A)$ fault in another basic stacking pattern. This is discussed in detail later. An example of the shifts in the decision parameters by the additional auxiliary parameters is shown in the table at the bottom of Figure S 2.

Integrating faults in an $(A\gamma B)(B\alpha C)(C\beta A)$ pattern by $(A\gamma B)(A\gamma B)$ type layer stacking requires an additional algorithm integrated in the first two ones (Figure S 3). An $(A\gamma B)(A\gamma B)$ type fault means that a layer is followed by another having the same position, i.e. $(A\gamma B)$ position layer is followed by $(A\gamma B)$ position, $(B\alpha C)$ position by $(B\alpha C)$ position and so on. This is achieved by defining additional auxiliary parameters ($\mathbf{aa}_i, \mathbf{bb}_i, \mathbf{cc}_i$, Table S 5). The assignment of 0 or 1 to these parameters is determined by the preceding layer (\mathbf{kx}_{i-1}) and their influence on the parameters of decision (\mathbf{deca}_i and \mathbf{decb}_i) is realized analogous to algorithm (2) (Figure S 2).

A global parameter for the probability of an $(A\gamma B)(A\gamma B)$ -fault in the $(A\gamma B)(B\alpha C)(C\beta A)$ stacking (\mathbf{paa}) is introduced and connected to the sum of auxiliary parameters ($\mathbf{aa}_i, \mathbf{bb}_i, \mathbf{cc}_i$) by multiplication, as well. To merge the real probability of a shift in stacking pattern with the defined probability (\mathbf{paa}), the definition of scaling factors ($\mathbf{noraaa}, \mathbf{norbbb}, \mathbf{norabb}, \mathbf{noracc}, \mathbf{norbcc}$) was necessary. The products of \mathbf{pabc} and \mathbf{paa} with their relating auxiliary parameters

are added to 1. This sum determines the value of the decision parameters (**deca_i** and **decb_i**). While **pabc** manipulates the decision parameters in a way in which an (A γ B)(B α C)(C β A) stacking is generated with the probability 1.0, the product of **paa** and its related auxiliary parameters and scale factors counteracts the influence of **pabc**. Accordingly the probability that a layer *i* is followed by an (A γ B)(B α C)(C β A) type stacked layer is reduced by the value of **paa**. In addition the probability that the layer *i* is followed by an (A γ B)(A γ B)-type stacked layer equals the value of **paa**. A detailed example of the impact of algorithm (3) on probability of the stacking sequences is given in Table S 6.

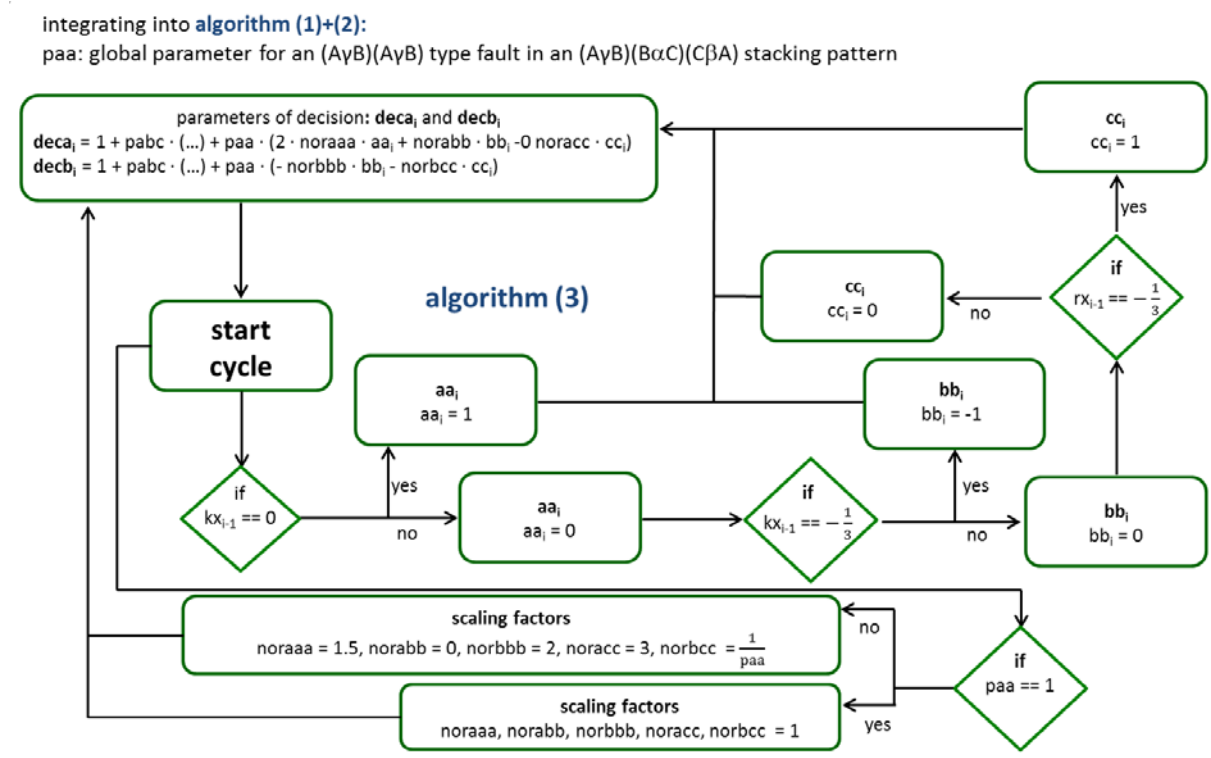


Figure S 3. Schematic representation of algorithm (3) for generation of shift to (A γ B)(A γ B) type stacking pattern with definite probability.

Table S 5. Overview of parameters used in algorithm (3).

parameter	description/ effect
paa	parameter for probability of (A γ B)(A γ B)-fault in (A γ B)(B α C)(C β A) type stacking, if paa < 1 parameter for (A γ B)(A γ B) type basis stacking, if paa = 1
aa_i	auxiliary parameter, partially counteracts the influence of ab_i on deca_i and decb_i → layer position (A γ B) is followed by (B α C) with the probability 1-paa, (A γ B) is followed by (A γ B) with the probability paa
bb_i	auxiliary parameter, partially counteracts the influence of bc_i on deca_i and decb_i → layer position (B α C) is followed by (C β A) with the probability 1-paa, (B α C) is followed by (B α C) with the probability paa
cc_i	auxiliary parameter, partially counteracts the influence of ca_i on deca_i → layer position (C β A) is followed by (A γ B) with the probability 1-paa, (C β A) is followed by (C β A) with the probability paa
noraaa	scaling factor, adjusts aa_i in calculation of deca_i to match calculated probability of an (A γ B)(A γ B) fault with the input of paa
norabb	scaling factor, adjusts bb_i in calculation of deca_i to match calculated probability of an (A γ B)(A γ B) fault with the input of paa
norbbb	scaling factor, adjusts bb_i in calculation of decb_i to match calculated probability of an (A γ B)(A γ B) fault with the input of paa
noracc	scaling factor, adjusts cc_i in calculation of deca_i to match calculated probability of an (A γ B)(A γ B) fault with the input of paa
norbcc	scaling factor, adjusts cc_i in calculation of decb_i to match calculated probability of an (A γ B)(A γ B) fault with the input of paa

Inversion of the stacking direction of the (A γ B)(B α C)(C β A) pattern can be simulated by shifts from (A γ B)(B α C)(C β A) to (C β A)(B α C)(A γ B) type stacking pattern. Those faults are implemented by an algorithm (Figure S 3) analogous to algorithm (3) (Figure S 4). Therefore a global parameter for the probability of a shift from (A γ B)(B α C)(C β A) to (C β A)(B α C)(A γ B) type stacking (**pab**) and additional auxiliary parameters (**ac_i**, **ba_i**, **cb_i**, Table S 7) are defined. Incorporating both (A γ B)(A γ B) and (C β A)(B α C)(A γ B) type faults in an (A γ B)(B α C)(C β A) stacking pattern requires further adjustment of the scaling factors. This is realized by algorithm (5) (Figure S 5). The adjusted scaling factors are fractions in which the shift probabilities (**paa** and **pab**) occur in the denominator. In order to prevent abnormal ends of TOPAS, caused by the division by 0, safety parameters (**saa**, **sab**, **saaab**, Table S 8) are defined. They are connected to the denominator by addition. When the shift probabilities (**paa**

and **pab**) assume values, that set the denominator to 0, 1 is assigned to the related safety parameter. In all other cases the safety parameter is set to 0 and therefore doesn't influence the scaling factor. A detailed example of the impact of algorithm (2)-(5) on the stacking probabilities is presented in Table S 9.

Table S 6. Detailed example of the application of algorithm 1-3 on the probabilities of generation of definite layers, using $pabc = 1.0$ and $paa = 0.2$, i.e. creating an $(A\gamma B)(A\gamma B)$ -faults in an $(A\gamma B)(B\alpha C)(C\beta A)$ stacking with the probability of 20 %.

definition: deca_i, decb_i	$\mathbf{deca}_i = 1 + pabc \cdot (\text{noraab} \cdot ab_i - \text{norabc} \cdot bc_i + 2 \cdot \text{noraca} \cdot ca_i) + paa \cdot (2 \cdot \text{noraaa} \cdot aa_i + \text{noabb} \cdot bb_i - \text{noracc} \cdot cc_i)$ $\mathbf{decb}_i = 1 + pabc \cdot (-\text{norbbc} \cdot bc_i - \text{norbab} \cdot ab_i) + paa \cdot (-\text{norbbb} \cdot bb_i - \text{norbcc} \cdot cc_i)$		
stacking probabilities	$p(kx_i = 0) = \frac{\mathbf{deca}_i}{3}$	$p(kx_i = -\frac{1}{3}) = \frac{3 - \mathbf{deca}_i \cdot \mathbf{decb}_i}{3 \cdot 2}$	$p(kx_i = \frac{1}{3}) = \frac{3 - \mathbf{deca}_i \cdot 2 - \mathbf{decb}_i}{3 \cdot 2}$
layer no.	deca_i , decb_i	p(kx = 0; 1/3, -1/3)	p(transition from → to)
<i>i - 1</i>	-	$kx_{i-1} = 0 := \text{start}$	$p((A\gamma B) \rightarrow (A\gamma B)) = 20.0 \%$
<i>i</i>	$\mathbf{deca}_i = 1 + 1 \cdot (-1 \cdot 1 - 1 \cdot 0 + 2 \cdot 1 \cdot 0) + 0.2 \cdot (2 \cdot 1.5 \cdot 1 + 0 \cdot 0 - 3 \cdot 0) = \mathbf{0.6}$	$p(kx_i = 0) = 0.20$ $p(kx_i = -\frac{1}{3}) = 0.80$ $p(kx_i = \frac{1}{3}) = 0.00$	$p(A\gamma B) \rightarrow (B\alpha C) = 80.0 \%$ $p(A\gamma B) \rightarrow (C\beta A) = 0.0 \%$
	$\mathbf{decb}_i = 1 + 1 \cdot (-1 \cdot 0 - (-1 \cdot 1)) + 0.2 \cdot (-2 \cdot 0 - \frac{1}{0.2} \cdot 0) = \mathbf{2}$		$p((B\alpha C) \rightarrow A\gamma B) = 0.0 \%$
<i>i + 1</i>	$\mathbf{deca}_i = 1 + 1 \cdot (1 \cdot 0 - 1 \cdot 1 + 2 \cdot 1 \cdot 0) + 0.2 \cdot (2 \cdot 1.5 \cdot 0 + 0 \cdot (-1) - 3 \cdot 0) = \mathbf{0.0}$	$p(kx_i = 0) = 0.00$ $p(kx_i = -\frac{1}{3}) = 0.20$ $p(kx_i = \frac{1}{3}) = 0.80$	$p((B\alpha C) \rightarrow (B\alpha C)) = 20.0 \%$ $p((B\alpha C) \rightarrow (C\beta A)) = 80.0 \%$
	$\mathbf{decb}_i = 1 + 1 \cdot (-1 \cdot 1 - 1 \cdot 0) + 0.2 \cdot (-2 \cdot (-1) - \frac{1}{0.2} \cdot 0) = \mathbf{0.4}$		$p((C\beta A) \rightarrow (A\gamma B)) = 80.0 \%$
<i>i + 2</i>	$\mathbf{deca}_i = 1 + 1 \cdot (1 \cdot 0 - 1 \cdot 0 + 2 \cdot 1 \cdot 1) + 0.2 \cdot (2 \cdot 1.5 \cdot 0 + 0 \cdot 0 - 3 \cdot 1) = \mathbf{2.4}$	$p(kx_i = 0) = 0.80$ $p(kx_i = -\frac{1}{3}) = 0.00$ $p(kx_i = \frac{1}{3}) = 0.20$	$p((C\beta A) \rightarrow (B\alpha C)) = 0.0 \%$ $p((C\beta A) \rightarrow (C\beta A)) = 20.0 \%$
	$\mathbf{decb}_i = 1 + 1 \cdot (-1 \cdot 0 - 1 \cdot 0) + 0.2 \cdot (-2 \cdot 0 - \frac{1}{0.2} \cdot 1) = \mathbf{0.0}$		

integrating into **algorithm (1)+(2)+(3)**:

pab: global parameter for shift from $(A\gamma B)(B\alpha C)(C\beta A)$ to $(C\beta A)(B\alpha C)(A\gamma B)$ type stacking (stacking direction change)

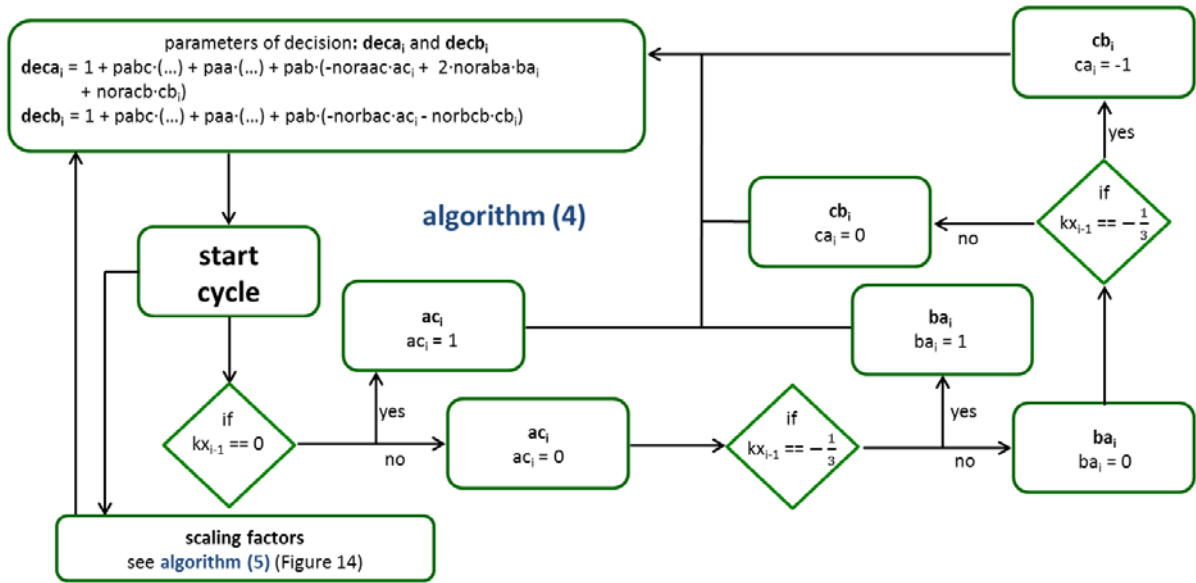


Figure S 4. Schematic representation of algorithm (4) for generation of $(C\beta A)(B\alpha C)(A\gamma B)$ type faults in $(A\gamma B)(B\alpha C)(C\beta A)$ stacking pattern (change in stacking direction) with definite probability.

Table S 7. Overview of parameters used in algorithm (4).

parameter	description/ effect
pab	parameter for probability of $(C\beta A)(B\alpha C)(A\gamma B)$ -fault in $(A\gamma B)(B\alpha C)(C\beta A)$ type stacking, if pab < 1 parameter for $(C\beta A)(B\alpha C)(A\gamma B)$ type basis stacking, if pab = 1
ac_i	auxiliary parameter, partially counteracts the influence of ab_i and aa_i on deca_i and decb_i → layer position $(A\gamma B)$ is followed by $(B\alpha C)$ with the probability $1-paa-pab$, $(A\gamma B)$ is followed by $(A\gamma B)$ with the probability paa or followed by $(C\beta A)$ with the probability pab
ba_i	auxiliary parameter, partially counteracts the influence of bc_i and bb_i on deca_i and decb_i → layer position $(B\alpha C)$ is followed by $(C\beta A)$ with the probability $1-paa-pab$, $(B\alpha C)$ is followed by $(B\alpha C)$ with the probability paa or followed by $(C\beta A)$ with the probability pab
cb_i	auxiliary parameter, partially counteracts the influence of ca_i and cc_i on deca_i → layer position $(C\beta A)$ is followed by $(A\gamma B)$ with the probability $1-paa-pab$, $(C\beta A)$ is followed by $(C\beta A)$ with the probability paa or followed by $(B\alpha C)$ with the probability pab
noraac	scaling factor, adjusts ac_i in calculation of deca_i to match calculated probability of an $(C\beta A)(B\alpha C)(A\gamma B)$ fault with the input of pab
noraba	scaling factor, adjusts ba_i in calculation of deca_i to match calculated probability of an $(C\beta A)(B\alpha C)(A\gamma B)$ fault with the input of pab

noracb	scaling factor, adjusts cb_i in calculation of deca_i to match calculated probability of an (CβA)(BαC)(AγB) fault with the input of pab
norbac	scaling factor, adjusts ac_i in calculation of decb_i to match calculated probability of an (CβA)(BαC)(AγB) fault with the input of pab
norbcb	scaling factor, adjusts cb_i in calculation of decb_i to match calculated probability of an (CβA)(BαC)(AγB) fault with the input of pab

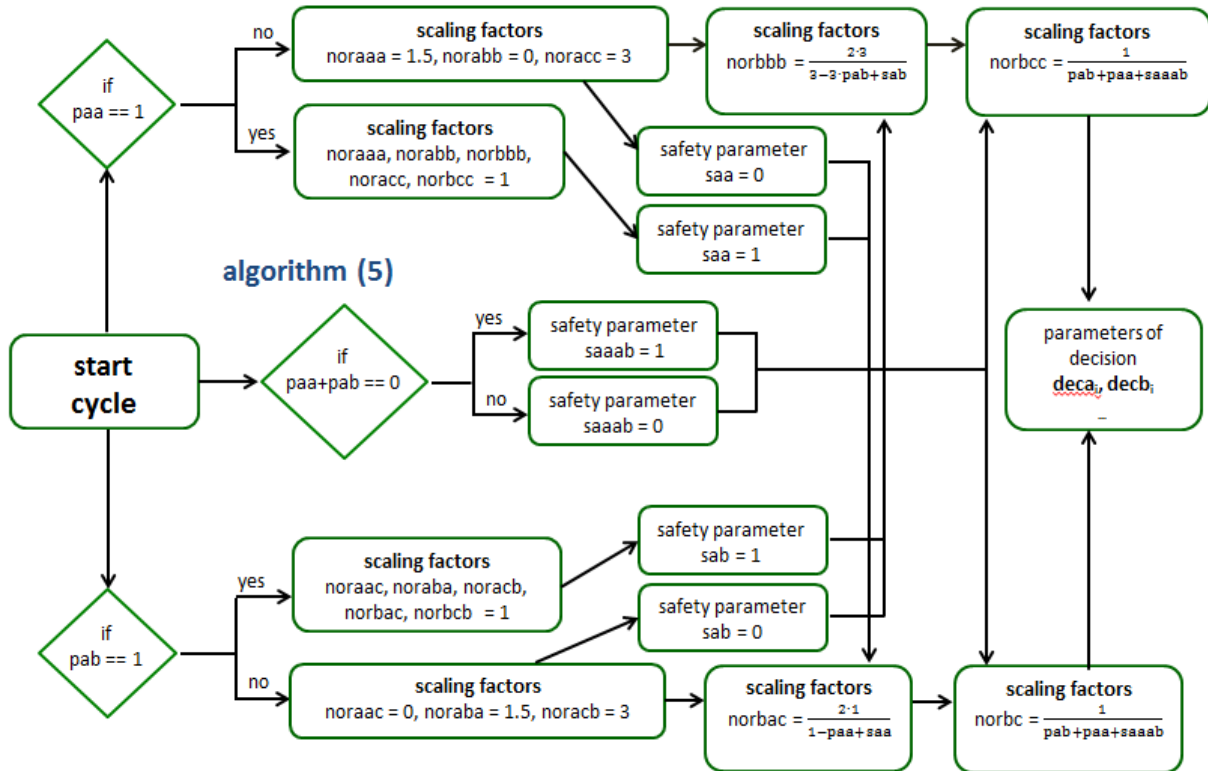


Figure S 5. Schematic representation of algorithm (5) for generation of scaling factors for two types of faults ((AγB)(AγB)- and (CβA)(BαC)(AγB)-type) in an (AγB)(BαC)(CβA) stacking pattern and safety parameters that prevents abnormal end of TOPAS by division by 0.

Table S 8. Overview of parameters used in algorithm (5).

parameter	description/ effect
saaab	safety parameter, prevents division by 0 in calculation of norbcc and norbc
sab	safety parameter, prevents division by 0 in calculation of norbbb
saa	safety parameter, prevents division by 0 in calculation of norbac

Table S 9. Detailed example of the application of algorithm 1-5 on the probabilities of generation of definite layers, using $p_{abc} = 1.0$, $p_{aa} = 0.2$, $p_{ab} 0.05$, i.e. creating a $(A\gamma B)(A\gamma B)$ type (probability of 20 %) and $(C\beta A)(B\alpha C)(A\gamma B)$ type faults (probability of 5 %) in an $(A\gamma B)(B\alpha C)(C\beta A)$ stacking.

definition:	$\mathbf{deca}_i = 1 + p_{abc} \cdot (\text{noraab} \cdot ab_i - \text{norabc} \cdot bc_i + 2 \cdot \text{noraca} \cdot ca_i) + p_{aa} \cdot (2 \cdot \text{noraaa} \cdot aa_i + \text{noabb} \cdot bb_i - \text{noracc} \cdot cc_i) + p_{ab} \cdot (-\text{noraac} \cdot ac_i + 2 \cdot \text{norba} \cdot ba_i + \text{noracb} \cdot cb_i)$		
$\mathbf{deca}_i,$ \mathbf{dec}_i	$\mathbf{dec}_i = 1 + p_{abc} \cdot (-\text{norbbc} \cdot bc_i - \text{norbab} \cdot ab_i) + p_{aa} \cdot (-\text{norbbb} \cdot bb_i - \text{norbcc} \cdot cc_i) + p_{ab} \cdot (-\text{norbac} \cdot ac_i - \text{norbc}_i \cdot cb_i)$		
stacking probabilities	$p(kx_i = 0) = \frac{\mathbf{deca}_i}{3}$	$p(kx_i = -\frac{1}{3}) = \frac{3 - \mathbf{deca}_i}{3} \cdot \frac{\mathbf{dec}_i}{2}$	$p(kx_i = \frac{1}{3}) = \frac{3 - \mathbf{deca}_i}{3} \cdot \frac{2 - \mathbf{dec}_i}{2}$
layer no.	$\mathbf{deca}_i, \mathbf{dec}_i$	$p(kx = 0; \frac{1}{3}, -\frac{1}{3})$	p(transition from \rightarrow to)
$i - 1$	-	$kx_{i-1} = 0 := \text{start}$	$p((A\gamma B) \rightarrow (A\gamma B)) = 20.0 \%$
i	$\mathbf{deca}_i = 1 + 1 \cdot (-1 \cdot 1 - 1 \cdot 0 + 2 \cdot 1 \cdot 0) + 0.20 \cdot (2 \cdot 1.5 \cdot 1 + 0 \cdot 0 - 3 \cdot 0) + 0.05 \cdot (-0 \cdot 1 + 2 \cdot 1.5 \cdot 0 - 3 \cdot 0) = \mathbf{0.6}$	$p(kx_i = 0) = 0.20$ $p(kx_i = -\frac{1}{3}) = 0.75$ $p(kx_i = \frac{1}{3}) = 0.05$	$p((A\gamma B) \rightarrow (B\alpha C)) = 75.0 \%$ $p((A\gamma B) \rightarrow (C\beta A)) = 5.0 \%$
	$\mathbf{dec}_i = 1 + 1 \cdot (-1 \cdot 0 - (-1 \cdot 1)) + 0.20 \cdot (-2 \cdot 0 \cdot \frac{3}{3 - 3 \cdot 0.05} - 0 \cdot \frac{1}{0.05 + 0.20}) + 0.05 \cdot (-2 \cdot 1 \cdot \frac{1}{1 - 0.20} - 0 \cdot \frac{1}{0.05 + 0.20}) = \mathbf{1.875}$		$p((B\alpha C) \rightarrow (A\gamma B)) = 5.0 \%$ $p((B\alpha C) \rightarrow (B\alpha C)) = 20.0 \%$
$i + 1$	$\mathbf{deca}_i = 1 + 1 \cdot (1 \cdot 0 - 1 \cdot 1 + 2 \cdot 1 \cdot 0) + 0.20 \cdot (2 \cdot 1.5 \cdot 0 + 0 \cdot (-1) - 3 \cdot 0) + 0.05 \cdot (-0 \cdot 0 + 2 \cdot 1.5 \cdot 1 - 3 \cdot 0) = \mathbf{0.15}$	$p(kx_i = 0) = 0.05$ $p(kx_i = -\frac{1}{3}) = 0.20$ $p(kx_i = \frac{1}{3}) = 0.75$	$p((B\alpha C) \rightarrow (C\beta A)) = 75.0 \%$
	$\mathbf{dec}_i = 1 + 1 \cdot (-1 \cdot 1 - 1 \cdot 0) + 0.20 \cdot (-2 \cdot (-1) \cdot \frac{3}{3 - 3 \cdot 0.05} - 0 \cdot \frac{1}{0.05 + 0.20}) + 0.05 \cdot (-2 \cdot 0 \cdot \frac{1}{1 - 0.20} - 0 \cdot \frac{1}{0.05 + 0.20}) = \mathbf{0.42}$		$p((C\beta A) \rightarrow (A\gamma B)) = 75.0 \%$ $p((C\beta A) \rightarrow (B\alpha C)) = 5.0 \%$ $p((C\beta A) \rightarrow (C\beta A)) = 20.0 \%$
$i + 2$	$\mathbf{deca}_i = 1 + 1 \cdot (1 \cdot 0 - 1 \cdot 0 + 2 \cdot 1 \cdot 1) + 0.20 \cdot (2 \cdot 1.5 \cdot 0 + 0 \cdot 0 - 3 \cdot 1) + 0.05 \cdot (-0 \cdot 0 + 2 \cdot 1.5 \cdot 0 - 3 \cdot 1) = \mathbf{2.25}$	$p(kx_i = 0) = 0.75$ $p(kx_i = -\frac{1}{3}) = 0.05$ $p(kx_i = \frac{1}{3}) = 0.20$	
	$\mathbf{dec}_i = 1 + 1 \cdot (-1 \cdot 0 - 1 \cdot 0) + 0.20 \cdot (-2 \cdot 0 \cdot \frac{3}{3 - 3 \cdot 0.05} - 1 \cdot \frac{1}{0.05 + 0.20}) + 0.05 \cdot (-2 \cdot 0 \cdot \frac{1}{1 - 0.20} - (-1) \cdot \frac{1}{0.05 + 0.20}) = \mathbf{0.4}$		

Both $(A\gamma B)(A\gamma B)$ and $(C\beta A)(B\alpha C)(A\gamma B)$ faults in an $(A\gamma B)(B\alpha C)(C\beta A)$ basis stacking pattern can be generated. In an analogous way, the $(A\gamma B)(A\gamma B)$ or $(C\beta A)(B\alpha C)(A\gamma B)$ pattern can be used as basis stacking. Accordingly stacking in an $(A\gamma B)(B\alpha C)(C\beta A)$ pattern will now account as a shift. Therefore the value of 1 is assigned to **paa** or **pab**. In consequence for every scaling factor of each pattern, multiple parameters have to be defined, which can be switched on and off depending on what is the current basic pattern. Hence each scaling factor for a definite pattern is the sum of three components, one for each basis pattern. Every summand is multiplied with an additional switch parameter (**ssabc**, **ssaa**, ..., Table S 10). Each of these switch parameter is set to 1 by a definite basis packing; the remaining two ones are assigned to 0. Therefore the proper component for a defined basis pattern is switched on and the other ones are switched off (algorithm (6), Figure S 6). Detailed example of stacking probabilities in faulted $(A\gamma B)(A\gamma B)$ and $(C\beta A)(B\alpha C)(A\gamma B)$ basis stacking can be found in Table S 11, Table S 12.

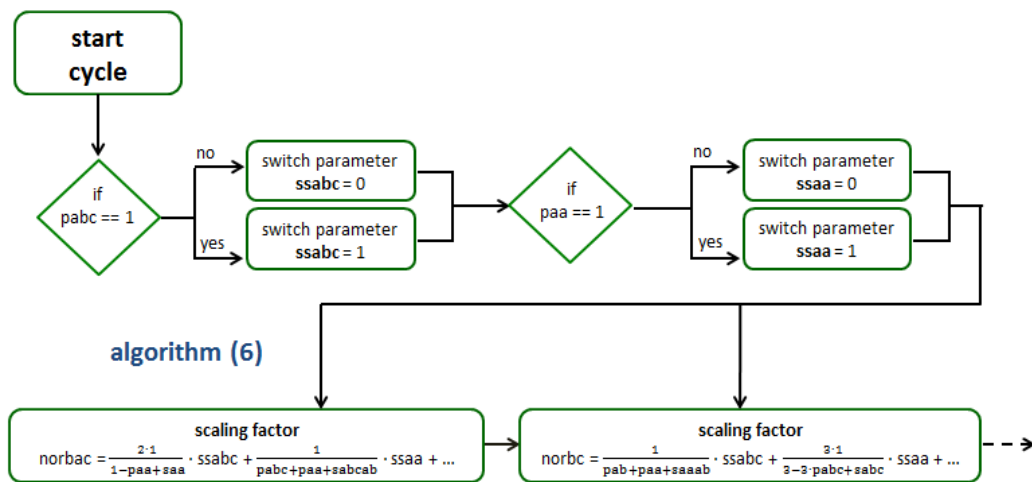


Figure S 6. Excerpt of algorithm (6) for switching of scaling factors in dependence on the basis pattern.

Table S 10. Overview of parameters used in algorithm (6).

parameter	description/ effect
ssabc	switch parameter, switching sets of scaling factors for $(A\gamma B)(B\alpha C)(C\beta A)$ base stacking or $(A\gamma B)(B\alpha C)(C\beta A)$ faults in other base stackings
ssaa	switch parameter, switching sets of scaling factors for $(A\gamma B)(A\gamma B)$ base stacking or $(A\gamma B)(A\gamma B)$ faults in other base stackings

Table S 11. Detailed example of the application of algorithm 1-6 on the probabilities of generation of definite layers, using $p_{abc} = 0.15$, $p_{aa} = 1.00$, $p_{ab} = 0.05$, i.e. creating a $(A\gamma B)(B\alpha C)(C\beta A)$ type (probability of 15 %) and $(C\beta A)(B\alpha C)(A\gamma B)$ type faults (probability of 5 %) in an $(A\gamma B)(A\gamma B)$ stacking.

definition:	$\text{deca}_i = 1 + p_{abc} \cdot (\text{noraab} \cdot \text{ab}_i - \text{norabc} \cdot \text{bc}_i + 2 \cdot \text{noraca} \cdot \text{ca}_i) + p_{aa} \cdot (2 \cdot \text{noraaa} \cdot \text{aa}_i + \text{noabb} \cdot \text{bb}_i - \text{noracc} \cdot \text{cc}_i) + p_{ab} \cdot (-\text{noraac} \cdot \text{ac}_i + 2 \cdot \text{norba} \cdot \text{ba}_i + \text{noracb} \cdot \text{cb}_i)$ $\text{decb}_i = 1 + p_{abc} \cdot (-\text{norbbc} \cdot \text{bc}_i - \text{norbab} \cdot \text{ab}_i) + p_{aa} \cdot (-\text{norbbb} \cdot \text{bb}_i - \text{norbcc} \cdot \text{cc}_i) + p_{ab} \cdot (-\text{norbac} \cdot \text{ac}_i - \text{norbc b} \cdot \text{cb}_i)$		
stacking probabilities	$p(kx_i = 0) = \frac{\text{deca}_i}{3}$	$p(kx_i = -\frac{1}{3}) = \frac{3 - \text{deca}_i}{3} \cdot \frac{\text{decb}_i}{2}$	$p(kx_i = \frac{1}{3}) = \frac{3 - \text{deca}_i}{3} \cdot \frac{2 - \text{decb}_i}{2}$
layer no.	$\text{deca}_i, \text{decb}_i$	$p(kx = 0; \frac{1}{3}, -\frac{1}{3})$	$p(\text{transition from } \rightarrow \text{ to})$
$i - 1$	-	$kx_{i-1} = 0 := \text{start}$	
i	$\text{deca}_i = 1 + 0.15 \cdot (-1 \cdot 3 - 0 \cdot 0 + 2 \cdot 2 \cdot 0) + 1 \cdot (2 \cdot 1 + 1 \cdot 0 - 1 \cdot 0) + 0.05 \cdot (-0 \cdot 1 + 2 \cdot 1.5 \cdot 0 - 3 \cdot 0) = \mathbf{2.4}$	$p(kx_i = 0) = 0.80$ $p(kx_i = -\frac{1}{3}) = 0.15$ $p(kx_i = \frac{1}{3}) = 0.05$	$p((A\gamma B) \rightarrow (A\gamma B)) = 80.0 \%$ $p((A\gamma B) \rightarrow (B\alpha C)) = 15.0 \%$ $p((A\gamma B) \rightarrow (C\beta A)) = 5.0 \%$
	$\text{decb}_i = 1 + 0.15 \cdot (-\frac{2}{1-0.05} \cdot 0 - (-1 \cdot \frac{1}{0.15+0.05})) + 1 \cdot (-1 \cdot 0 \cdot (-1) \cdot 0) + 0.05 \cdot (-1 \cdot \frac{1}{0.15+0.05} - 0 \cdot \frac{2 \cdot 3}{3-3 \cdot 0.15}) = \mathbf{1.5}$		
$i - 1$	-	$kx_{i-1} = \frac{1}{3} := \text{start}$	
i	$\text{deca}_i = 1 + 0.15 \cdot (3 \cdot 0 - 0 \cdot 1 + 2 \cdot 2 \cdot 0) + 1 \cdot (2 \cdot 0 + 1 \cdot (-1) - 1 \cdot 0) + 0.05 \cdot (-0 \cdot 0 + 2 \cdot 1.5 \cdot 1 - 3 \cdot 0) = \mathbf{0.15}$	$p(kx_i = 0) = 0.05$ $p(kx_i = -\frac{1}{3}) = 0.80$ $p(kx_i = \frac{1}{3}) = 0.15$	$p((B\alpha C) \rightarrow (A\gamma B)) = 5.0 \%$ $p((B\alpha C) \rightarrow (B\alpha C)) = 80.0 \%$ $p((B\alpha C) \rightarrow (C\beta A)) = 15.0 \%$
	$\text{decb}_i = 1 + 0.15 \cdot (-\frac{2}{1-0.05} \cdot 1 - \frac{1}{0.15+0.05} \cdot 0) + 1 \cdot (-1 \cdot (-1) - 1 \cdot 0) + 0.05 \cdot (-\frac{1}{0.15+0.05} \cdot 0 - 0 \cdot \frac{2 \cdot 3}{3-3 \cdot 0.15}) = \mathbf{1.684}$		
$i - 1$	-	$kx_{i-1} = \frac{1}{3} := \text{start}$	
i	$\text{deca}_i = 1 + 0.15 \cdot (3 \cdot 0 - 0 \cdot 0 + 2 \cdot 2 \cdot 1) + 1 \cdot (2 \cdot 0 + 1 \cdot 0 - 1 \cdot 1) + 0.05 \cdot (-0 \cdot 0 + 2 \cdot 1.5 \cdot 0 - 3 \cdot 1) = \mathbf{0.45}$	$p(kx_i = 0) = 0.15$ $p(kx_i = -\frac{1}{3}) = 0.05$ $p(kx_i = \frac{1}{3}) = 0.80$	$p((C\beta A) \rightarrow (A\gamma B)) = 15.0 \%$ $p((C\beta A) \rightarrow (B\alpha C)) = 5.0 \%$ $p((C\beta A) \rightarrow (C\beta A)) = 80.0 \%$
	$\text{decb}_i = 1 + 0.15 \cdot (-\frac{2}{1-0.05} \cdot 0 - \frac{1}{0.15+0.05} \cdot 0) + 1 \cdot (-1 \cdot 0 - 1 \cdot 1) + 0.05 \cdot (-\frac{1}{0.15+0.05} \cdot 0 - (-1) \cdot \frac{2 \cdot 3}{3-3 \cdot 0.15}) = \mathbf{0.118}$		

Table S 12. Detailed example of the application of algorithm 1-6 on the probabilities of generation of definite layers, using $p_{abc} = 0.20$, $p_{aa} = 0.10$, $p_{ab} = 1.00$, i.e. creating a $(A\gamma B)(B\alpha C)(C\beta A)$ type (probability of 20 %) and $(A\gamma B)(A\gamma B)$ type faults (probability of 10 %) in an $(C\beta A)(B\alpha C)(A\gamma B)$ stacking.

definition:	$\mathbf{deca}_i = 1 + p_{abc} \cdot (\text{noraab} \cdot ab_i - \text{norabc} \cdot bc_i + 2 \cdot \text{noraca} \cdot ca_i) + p_{aa} \cdot (2 \cdot \text{noraaa} \cdot aa_i + \text{noabb} \cdot bb_i - \text{noracc} \cdot cc_i) + p_{ab} \cdot (-\text{noraac} \cdot ac_i + 2 \cdot \text{norba} \cdot ba_i + \text{noracb} \cdot cb_i)$ $\mathbf{decb}_i = 1 + p_{abc} \cdot (-\text{norbbc} \cdot bc_i - \text{norbab} \cdot ab_i) + p_{aa} \cdot (-\text{norbbb} \cdot bb_i - \text{norbcc} \cdot cc_i) + p_{ab} \cdot (-\text{norbac} \cdot ac_i - \text{norbc} \cdot cb_i)$		
stacking probabilities	$p(kx_i = 0) = \frac{\mathbf{deca}_i}{3}$	$p(kx_i = -\frac{1}{3}) = \frac{3 - \mathbf{deca}_i}{3} \cdot \frac{\mathbf{decb}_i}{2}$	$p(kx_i = \frac{1}{3}) = \frac{3 - \mathbf{deca}_i}{3} \cdot \frac{2 - \mathbf{decb}_i}{2}$
layer no.	$\mathbf{deca}_i, \mathbf{decb}_i$	$p(kx = 0; \frac{1}{3}, -\frac{1}{3})$	$p(\text{transition from } \rightarrow \text{ to})$
$i - 1$	-	$kx_{i-1} = 0 := \text{start}$	$p((A\gamma B) \rightarrow (A\gamma B)) = 10.0 \%$
i	$\mathbf{deca}_i = 1 + 0.2 \cdot (-1 \cdot 0 - 3 \cdot 0 + 3 \cdot 0) + 0.1 \cdot (2 \cdot 1.5 \cdot 1 + 3 \cdot 0 - 0 \cdot 0) + 1.0 \cdot (-1 \cdot 1 + 2 \cdot 1 \cdot 0 - 1 \cdot 0) = \mathbf{0.3}$	$p(kx_i = 0) = 0.10$	$p((A\gamma B) \rightarrow (B\alpha C)) = 20.0 \%$
	$\mathbf{decb}_i = 1 + 0.2 \cdot (-\frac{1}{0.2+0.1} \cdot 0 - (-1 \cdot \frac{2 \cdot 3}{3-3 \cdot 0.10})) + 0.1 \cdot (\frac{1}{0.2+0.1} \cdot 0 \cdot (-1) \cdot 0) + 1.0 \cdot (-1 \cdot 1 - 0 \cdot 1) = \mathbf{0.444}$	$p(kx_i = -\frac{1}{3}) = 0.20$ $p(kx_i = \frac{1}{3}) = 0.70$	$p((A\gamma B) \rightarrow (C\beta A)) = 70.0 \%$
$i - 1$	-	$kx_{i-1} = \frac{1}{3} := \text{start}$	$p((B\alpha C) \rightarrow (A\gamma B)) = 70.0 \%$
i	$\mathbf{deca}_i = 1 + 0.2 \cdot (0 \cdot 0 - 3 \cdot 1 + 3 \cdot 0) + 0.1 \cdot (2 \cdot 1.5 \cdot 0 + 3 \cdot (-1) - 0 \cdot 0) + 1.0 \cdot (-0 \cdot 1 + 2 \cdot 1 \cdot 1 - 1 \cdot 0) = \mathbf{2.1}$	$p(kx_i = 0) = 0.70$	$p((B\alpha C) \rightarrow (B\alpha C)) = 10.0 \%$
	$\mathbf{decb}_i = 1 + 0.2 \cdot (-\frac{1}{0.2+0.1} \cdot 1 - \frac{2 \cdot 3}{3-3 \cdot 0.10} \cdot 0) + 0.1 \cdot (\frac{1}{0.2+0.1} \cdot (-1) - 1 \cdot 0) + 1.0 \cdot (-1 \cdot 0 - 0 \cdot 1) = \mathbf{0.666}$	$p(kx_i = -\frac{1}{3}) = 0.10$ $p(kx_i = \frac{1}{3}) = 0.20$	$p((B\alpha C) \rightarrow (C\beta A)) = 20.0 \%$
$i - 1$	-	$kx_{i-1} = \frac{1}{3} := \text{start}$	$p((C\beta A) \rightarrow (A\gamma B)) = 20.0 \%$
i	$\mathbf{deca}_i = 1 + 0.2 \cdot (0 \cdot 0 - 3 \cdot 0 + 3 \cdot 1) + 0.1 \cdot (2 \cdot 1.5 \cdot 0 + 3 \cdot 0 - 0 \cdot 1) + 1.0 \cdot (-0 \cdot 1 + 2 \cdot 1 \cdot 0 - 1 \cdot 1) = \mathbf{0.6}$	$p(kx_i = 0) = 0.20$	$p((C\beta A) \rightarrow (B\alpha C)) = 70.0 \%$
	$\mathbf{decb}_i = 1 + 0.2 \cdot (-\frac{1}{0.2+0.1} \cdot 0 - \frac{2 \cdot 3}{3-3 \cdot 0.10} \cdot 0) + 0.1 \cdot (\frac{1}{0.2+0.1} \cdot 0 - 1 \cdot \frac{2}{1-0.2}) + 1.0 \cdot (-1 \cdot 0 - (-1) \cdot 1) = \mathbf{1.75}$	$p(kx_i = -\frac{1}{3}) = 0.70$ $p(kx_i = \frac{1}{3}) = 0.10$	$p((C\beta A) \rightarrow (C\beta A)) = 10.0 \%$

By combining of algorithm (1)-(6) $(A\gamma B)(B\alpha C)(C\beta A)$, $(A\gamma B)(A\gamma B)$ or $(C\beta A)(B\alpha C)(A\gamma B)$ stackings can be used as basis pattern including faults of the other two types. In order to simulate all microstructure features we assumed and to perform a *DIFFaX* like recursion, **shifting** from one basis stacking type to the other is indispensable, e.g. **shifting** from $(A\gamma B)(B\alpha C)(C\beta A)$ to $(A\gamma B)(A\gamma B)$ basis stacking pattern. This is the only way to use stacking probabilities as defined in Table 4 for input. Thus the data input matches completely with the input used for the *DIFFaX* simulations. This requires additional algorithms that identify the current basic stacking pattern autonomously. Accordingly the algorithms has to “know” which probability to choose (P11, P21, P31) to stack the layer number i in an $(A\gamma B)(B\alpha C)(C\beta A)$ stacking pattern (Figure S 7). Therefore, whether stacking layer i in an $(A\gamma B)(B\alpha C)(C\beta A)$ pattern means that a current $(A\gamma B)(B\alpha C)(C\beta A)$ pattern is continued (P11), or an $(A\gamma B)(B\alpha C)(C\beta A)$ fault is generated in an $(A\gamma B)(A\gamma B)$ (P21) or in a $(C\beta A)(B\alpha C)(A\gamma B)$ type pattern (P31).

An automated detection of the current basic stacking pattern can be realized, when the stacking vectors of two layers, directly preceding $(i-1, i-2)$ layer i are taken into account. The basic stacking pattern can be identified bijectively by subtracting the stacking vectors of adjacent layers (eq. (2)). As the basic stacking patterns are represented by distinct sequences of stacking vectors (Table 9), this difference must be 0 for $(A\gamma B)(A\gamma B)$ stacking, $-\frac{1}{3}$ or $\frac{2}{3}$ for $(A\gamma B)(B\alpha C)(C\beta A)$ stacking and $\frac{1}{3}$ or $-\frac{2}{3}$ for $(C\beta A)(B\alpha C)(A\gamma B)$ type stacking.

$$\text{eq. (1)} \quad k_{x_{i-1}} - k_{x_{i-2}}$$

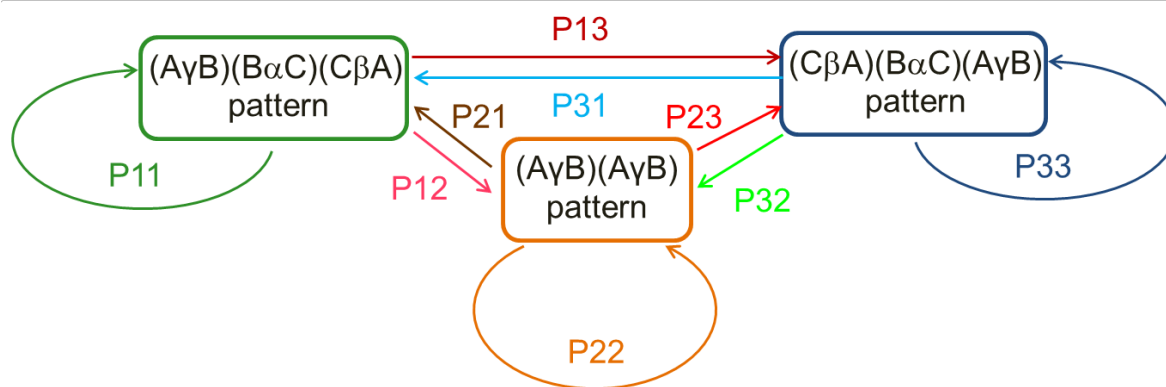


Figure S 7. Illustration of the basis stacking pattern and the stacking probabilities for continuation of a current pattern and shift to another stacking type.

Accordingly additional algorithms (7a - c) were implemented (Figure S 8-Figure S 10). Switch parameters were defined for each stacking type (e.g. **helper1pabc_i**, **helper2pabc_i**, **helper3pabc_i** for $(A\gamma B)(B\alpha C)(C\beta A)$ stacking, Table S 13). If the difference, $k_{x_{i-1}} - k_{x_{i-2}}$ matches to an $(A\gamma B)(B\alpha C)(C\beta A)$ stacking pattern, i.e. $k_{x_{i-1}} - k_{x_{i-2}} = -\frac{1}{3}$ or $\frac{2}{3}$, the value 0 will be assigned to one of the switch parameters (**helper1pabc_i** for $k_{x_{i-1}} - k_{x_{i-2}} = -\frac{1}{3}$, **helper2pabc_i** for $k_{x_{i-1}} - k_{x_{i-2}} = \frac{2}{3}$). These parameters are connected by multiplication (Figure S 8). If the product of **helper1pabc_i** and **helper2pabc_i** is 0, a third switch parameter (**helper3pabc_i**) will be assigned to 1. Then the local parameter **pabc_i** is set to 1, which means that the current basic stacking pattern is $(A\gamma B)(B\alpha C)(C\beta A)$ type. If the product of **helper1pabc_i** and **helper2pabc_i** is 0, which means that the current stacking pattern is not $(A\gamma B)(B\alpha C)(C\beta A)$ type, the third switch parameter (**helper3pabc_i**) will be assigned to 0. In consequence the local parameter **pabc_i** is set to P21 (shift from $(A\gamma B)(A\gamma B)$ to $(A\gamma B)(B\alpha C)(C\beta A)$ stacking) or P31 (shift from $(C\beta A)(B\alpha C)(A\gamma B)$ to $(A\gamma B)(B\alpha C)(C\beta A)$ stacking). The latter decision is determined by equivalent algorithms ((7b) and (c), Figure S 9, Table S 14, Figure S 10, Table S 12), for identifying the current basic stacking pattern either as $(A\gamma B)(A\gamma B)$ (algorithm (7b), Figure S 9) or $(C\beta A)(B\alpha C)(A\gamma B)$ type (algorithm (7c), Figure S 10).

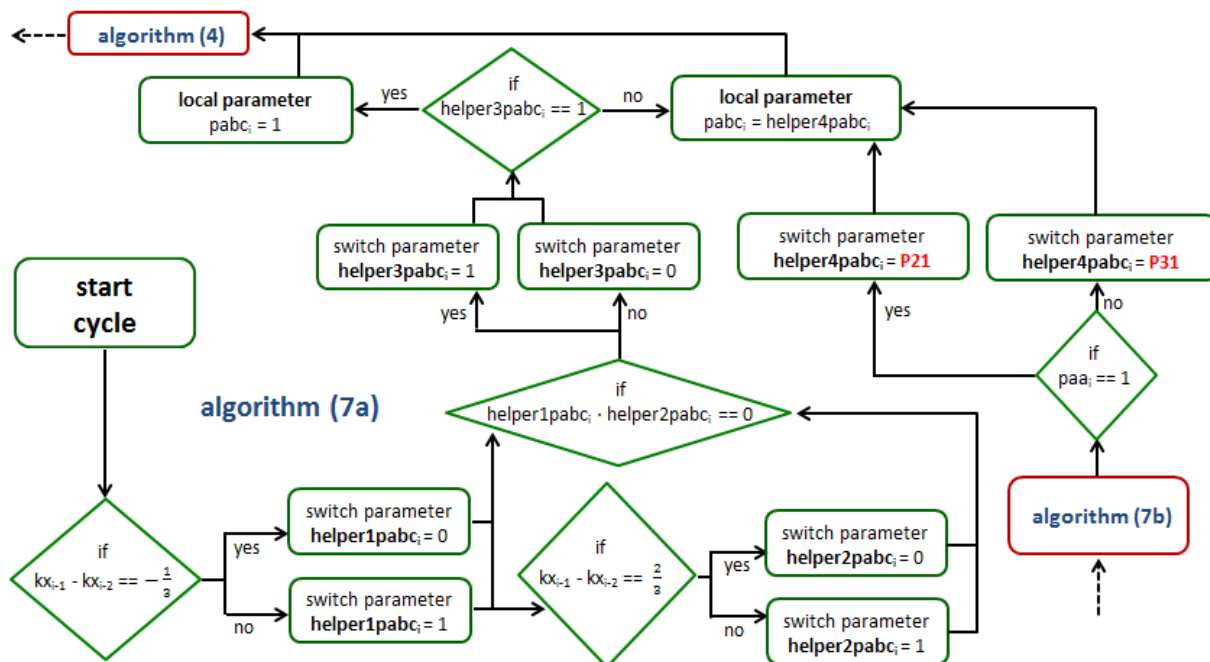


Figure S 8. Illustration of algorithm (7a) for automated identification of an $(A\gamma B)(B\alpha C)(C\beta A)$ type basis stacking and assignment of proper stacking probabilities.

Table S 13. Overview of parameters used in algorithm (7a).

parameter	description/ effect
helper1pabc_i	switch parameter, identification of $(A\gamma B)(B\alpha C)(C\beta A)$ as current base stacking pattern, is set to 0, if the two preceding layers show $(A\gamma B)(B\alpha C)$ or $(C\beta A)(A\gamma B)$ sequence, otherwise to 1
helper2pabc_i	switch parameter, identification of $(A\gamma B)(B\alpha C)(C\beta A)$ as current base stacking pattern, is set to 0, if the two preceding layers show $(B\alpha C)(C\beta A)$ sequence, otherwise to 1
helper3pabc_i	switch parameter, adjusts pabc_i either to 1.0 (in case of $(A\gamma B)(B\alpha C)(C\beta A)$ base stacking, helper1pabc_i or helper2pabc_i = 0) or to helper4pabc_i
helper4pabc_i	switch parameter, adjusts pabc_i either to P21 (for $(A\gamma B)(A\gamma B)$ base stacking) or to P31 (for $(C\beta A)(B\alpha C)(A\gamma B)$ base stacking)

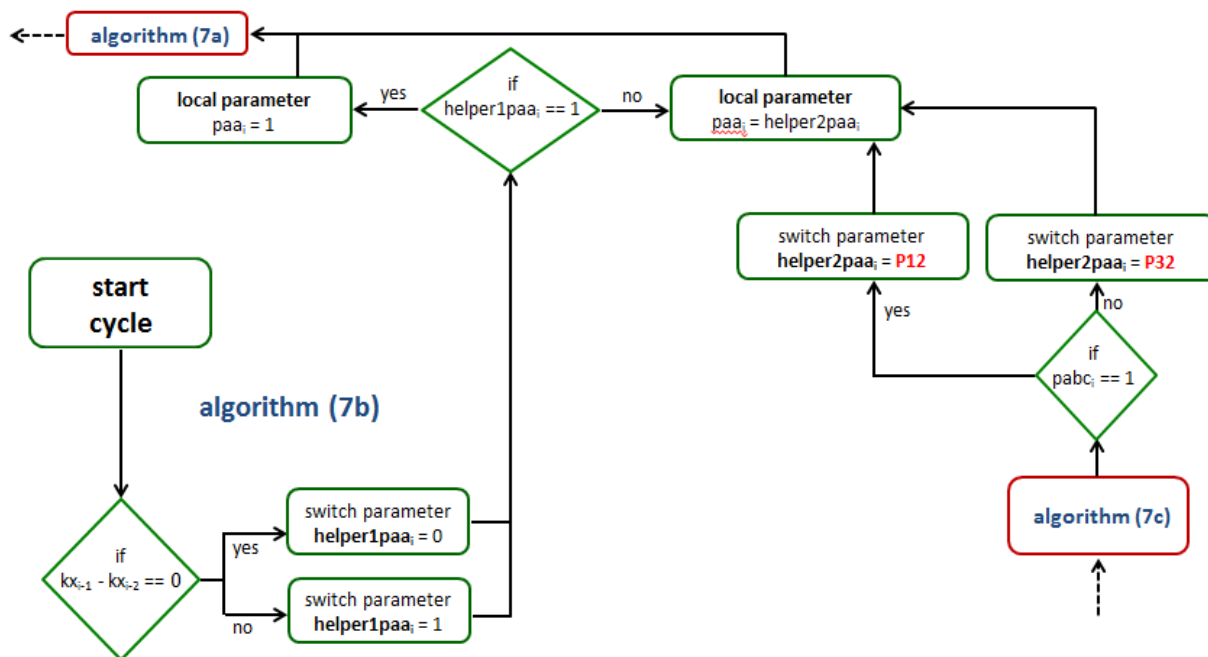


Figure S 9. Illustration of algorithm (7b) for automated identification of an $(A\gamma B)(A\gamma B)$ type basis stacking and assignment of proper stacking probabilities.

Table S 14. Overview of parameters used in algorithm (7b).

parameter	description/ effect
helper1paa_i	switch parameter, identification of $(A\gamma B)(A\gamma B)$ as current base stacking pattern, is set to 0, if the two preceding layers show $(A\gamma B)(A\gamma B)$, $(B\alpha C)(B\alpha C)$ or $(C\beta A)(C\beta A)$ sequence, otherwise to 1
helper2paa_i	switch parameter, adjusts paa_i either to P12 (for $(A\gamma B)(A\gamma B)$ base stacking) or to P32 (for $(C\beta A)(B\alpha C)(A\gamma B)$ base stacking)

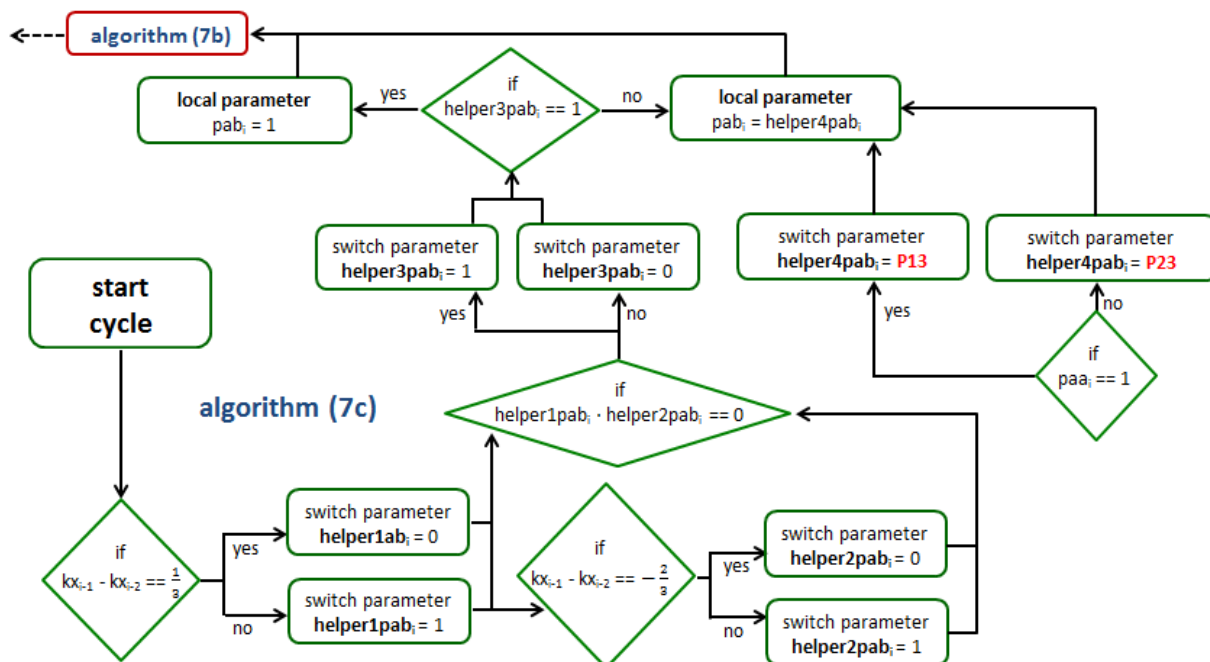


Figure S 10. Illustration of algorithm (7c) for automated identification of an $(C\beta A)(B\alpha C)(A\gamma B)$ type basis stacking and assignment of proper stacking probabilities.

Table S 15. Overview of parameters used in algorithm (7b).

parameter	description/ effect
helper1pab_i	switch parameter, identification of $(C\beta A)(B\alpha C)(A\gamma B)$ as current base stacking pattern, is set to 0, if the two preceding layers show $(B\alpha C)(A\gamma B)$ or $(A\gamma B)(C\beta A)$ sequence, otherwise to 1
helper2pab_i	switch parameter, identification of $(C\beta A)(B\alpha C)(A\gamma B)$ as current base stacking pattern, is set to 0, if the two preceding layers show $(C\beta A)(B\alpha C)$ sequence, otherwise to 1
helper3pab_i	switch parameter, adjusts pabc_i either to 1.0 (in case of $(A\gamma B)(B\alpha C)(C\beta A)$ base stacking, helper1pab_i or helper2pab_i = 0) or to helper4pab_i
helper4pab_i	switch parameter, adjusts pab_i either to P23 (for $(A\gamma B)(A\gamma B)$ base stacking) or to P13 (for $(A\gamma B)(B\alpha C)(C\beta A)$ base stacking)

Finally a start parameter was defined which sets the “fictive” layers no. -1 and -2 to $(A\gamma B)(A\gamma B)$ or $(A\gamma B)(B\alpha C)(C\beta A)$ stacking type, depending on the input by the user.

Appendix C. recursive routine for global optimization in TOPAS syntax

```
process_times
continue_after_convergence
temperature 1
temperature 1
temperature 1
temperature 1
temperature 1
temperature 1.2
iters 1000000
r_wp 8.03578364
'-----
#####
#####
INPUT STACKING PROBABILITIES
/*
MANUEL
What can be done with this INP-file?:
- The INP-file uses a super cell approach, the number of layers is set to 36 by default.
- The space group P1 MUST BE used!
- The stacking direction is c.
- The INP-file uses a simulated annealing as global optimization method.
- The position of each layer is described by a stacking vector with the origin at (0, 0, 0).
- Each stacking vector consists of a regular part ((AγB)-, (BαC)-, (CβA)- layerpositions), created by global
optimization and random part for modelling small, random
(turbostratic like) shifts in the stacking pattern.
- Set the initial basic stacking pattern by the parameter start to (AγB)(BαC)(CβA), (AγB)(AγB) or (CβA)(BαC)
(AγB) ((AγB)(BαC)(CβA) stacking with inverted stacking order).
- Probabilities for a shift of the stacking pattern can be defined using the parameters P12, P13, P21, P23, P31,
P32, values < 1 and >=0 must be assigned.
- At the beginning of each cycle a stacking sequence is created randomly using the probabilities for a shift of the
basic stacking pattern defined by
P12, P13, P21, P23, P31, P32.
- The random part of the stacking vector is refined at the end of each cycle.
- The parameter dts activates the refinement of small, random (turbostratic like) shifts in the stacking pattern. dts
= 1: activate dts = 0 deactivate
- The parameter limts defines the limits for refinement of the small, random (turbostratic like) shifts in the
stacking pattern.
*/
#####
#####
#####
#####
INPUT: STACKING PROBABILITIES REGULAR PART
'starting parameter: 1 for initial (AγB)(BαC)(CβA) basis stacking, 0 for initial (AγB)(AγB) basis stacking
prm !start 1
#####
#####
'Use values < 1, >= 0 for these parameters!
'(AγB)(BαC)(CβA)-stacking
prm !P12 0.06 'probability for a shift from (AγB)(BαC)(CβA) to (AγB)(AγB) stacking pattern
prm !P13 0.03 'probability for a shift from (AγB)(BαC)(CβA) to (CβA)(BαC) (AγB) stacking pattern
#####
#####
'(AγB)(AγB)-stacking
prm !P21 0.15 'probability for a shift from (AγB)(AγB) to (AγB)(BαC)(CβA) stacking pattern
prm !P23 0.03 'probability for a shift from (AγB)(AγB) to (CβA)(BαC)(AγB) stacking pattern
#####
#####
```

'(CβA)(BαC)(AγB)-stacking
prm !P31 0.65 'probability for a shift from CBA to (AγB)(BαC)(CβA) stacking pattern
prm !P32 0.35 'probability for a shift from CBA to (AγB)(AγB) stacking pattern

#####

INPUT: RANDOM PART
prm !limits 0.10 'limits for the refinement of the random part, must never exceed 1/6

prm !dts 1 'activation/ deactivation of the random part

#####

'CALCULATION AREA: DO ONLY MAKE CHANGES, WHEN YOU CHANGE THE NUMBER OF
LAYERS IN THE SUPERCELL!

#####

'RANDOM NUMBER CREATION FOR REGULAR STACKING SEQUENCE

prm stacka2 2.17752` val_on_continue = Rand(0, 3); prm stackb2 1.80529` val_on_continue = Rand(0, 2);
prm stacka3 1.81922` val_on_continue = Rand(0, 3); prm stackb3 1.50564` val_on_continue = Rand(0, 2);
prm stacka4 1.77381` val_on_continue = Rand(0, 3); prm stackb4 1.37563` val_on_continue = Rand(0, 2);
prm stacka5 0.83760` val_on_continue = Rand(0, 3); prm stackb5 0.06186` val_on_continue = Rand(0, 2);
prm stacka6 2.97311` val_on_continue = Rand(0, 3); prm stackb6 0.73980` val_on_continue = Rand(0, 2);
prm stacka7 0.95286` val_on_continue = Rand(0, 3); prm stackb7 1.47703` val_on_continue = Rand(0, 2);
prm stacka8 0.01736` val_on_continue = Rand(0, 3); prm stackb8 1.67906` val_on_continue = Rand(0, 2);
prm stacka9 0.63992` val_on_continue = Rand(0, 3); prm stackb9 1.17918` val_on_continue = Rand(0, 2);
prm stacka10 1.99267` val_on_continue = Rand(0, 3); prm stackb10 1.53606` val_on_continue = Rand(0, 2);
prm stacka11 2.20782` val_on_continue = Rand(0, 3); prm stackb11 0.09981` val_on_continue = Rand(0, 2);
prm stacka12 0.02741` val_on_continue = Rand(0, 3); prm stackb12 0.41108` val_on_continue = Rand(0, 2);
prm stacka13 0.93582` val_on_continue = Rand(0, 3); prm stackb13 1.49392` val_on_continue = Rand(0, 2);
prm stacka14 2.36760` val_on_continue = Rand(0, 3); prm stackb14 1.60632` val_on_continue = Rand(0, 2);
prm stacka15 0.08582` val_on_continue = Rand(0, 3); prm stackb15 1.00997` val_on_continue = Rand(0, 2);
prm stacka16 2.32483` val_on_continue = Rand(0, 3); prm stackb16 1.32192` val_on_continue = Rand(0, 2);
prm stacka17 0.78855` val_on_continue = Rand(0, 3); prm stackb17 0.01089` val_on_continue = Rand(0, 2);
prm stacka18 2.10192` val_on_continue = Rand(0, 3); prm stackb18 1.60096` val_on_continue = Rand(0, 2);
prm stacka19 2.80561` val_on_continue = Rand(0, 3); prm stackb19 0.14392` val_on_continue = Rand(0, 2);
prm stacka20 2.49083` val_on_continue = Rand(0, 3); prm stackb20 0.52482` val_on_continue = Rand(0, 2);
prm stacka21 1.16575` val_on_continue = Rand(0, 3); prm stackb21 1.16103` val_on_continue = Rand(0, 2);
prm stacka22 0.44782` val_on_continue = Rand(0, 3); prm stackb22 1.65872` val_on_continue = Rand(0, 2);
prm stacka23 2.67728` val_on_continue = Rand(0, 3); prm stackb23 1.98919` val_on_continue = Rand(0, 2);
prm stacka24 0.54246` val_on_continue = Rand(0, 3); prm stackb24 0.74651` val_on_continue = Rand(0, 2);
prm stacka25 2.31431` val_on_continue = Rand(0, 3); prm stackb25 1.05068` val_on_continue = Rand(0, 2);
prm stacka26 1.28105` val_on_continue = Rand(0, 3); prm stackb26 1.65662` val_on_continue = Rand(0, 2);
prm stacka27 0.47823` val_on_continue = Rand(0, 3); prm stackb27 1.83373` val_on_continue = Rand(0, 2);
prm stacka28 1.30713` val_on_continue = Rand(0, 3); prm stackb28 1.57684` val_on_continue = Rand(0, 2);
prm stacka29 1.13532` val_on_continue = Rand(0, 3); prm stackb29 1.71223` val_on_continue = Rand(0, 2);
prm stacka30 1.47475` val_on_continue = Rand(0, 3); prm stackb30 1.45998` val_on_continue = Rand(0, 2);
prm stacka31 2.82758` val_on_continue = Rand(0, 3); prm stackb31 0.01649` val_on_continue = Rand(0, 2);
prm stacka32 2.79802` val_on_continue = Rand(0, 3); prm stackb32 1.71892` val_on_continue = Rand(0, 2);
prm stacka33 0.55957` val_on_continue = Rand(0, 3); prm stackb33 1.35363` val_on_continue = Rand(0, 2);
prm stacka34 0.27516` val_on_continue = Rand(0, 3); prm stackb34 1.17228` val_on_continue = Rand(0, 2);

prm stacka35 1.56086` val_on_continue = Rand(0, 3); prm stackb35 0.17735` val_on_continue = Rand(0, 2);
prm stacka36 2.58850` val_on_continue = Rand(0, 3); prm stackb36 1.23937` val_on_continue = Rand(0, 2);

#####

'SELECTION OF THE STACKING VECTORS (REGULAR PART)

local kora2 = If (stacka2 <=deca2, 0, 1); local korb2 = If (stackb2 <=decb2, -1, 1);
local kora3 = If (stacka3 <=deca3, 0, 1); local korb3 = If (stackb3 <=decb3, -1, 1);
local kora4 = If (stacka4 <=deca4, 0, 1); local korb4 = If (stackb4 <=decb4, -1, 1);
local kora5 = If (stacka5 <=deca5, 0, 1); local korb5 = If (stackb5 <=decb5, -1, 1);
local kora6 = If (stacka6 <=deca6, 0, 1); local korb6 = If (stackb6 <=decb6, -1, 1);
local kora7 = If (stacka7 <=deca7, 0, 1); local korb7 = If (stackb7 <=decb7, -1, 1);
local kora8 = If (stacka8 <=deca8, 0, 1); local korb8 = If (stackb8 <=decb8, -1, 1);
local kora9 = If (stacka9 <=deca9, 0, 1); local korb9 = If (stackb9 <=decb9, -1, 1);
local kora10 = If (stacka10 <=deca10, 0, 1); local korb10 = If (stackb10 <=decb10, -1, 1);
local kora11 = If (stacka11 <=deca11, 0, 1); local korb11 = If (stackb11 <=decb11, -1, 1);
local kora12 = If (stacka12 <=deca12, 0, 1); local korb12 = If (stackb12 <=decb12, -1, 1);
local kora13 = If (stacka13 <=deca13, 0, 1); local korb13 = If (stackb13 <=decb13, -1, 1);
local kora14 = If (stacka14 <=deca14, 0, 1); local korb14 = If (stackb14 <=decb14, -1, 1);
local kora15 = If (stacka15 <=deca15, 0, 1); local korb15 = If (stackb15 <=decb15, -1, 1);
local kora16 = If (stacka16 <=deca16, 0, 1); local korb16 = If (stackb16 <=decb16, -1, 1);
local kora17 = If (stacka17 <=deca17, 0, 1); local korb17 = If (stackb17 <=decb17, -1, 1);
local kora18 = If (stacka18 <=deca18, 0, 1); local korb18 = If (stackb18 <=decb18, -1, 1);
local kora19 = If (stacka19 <=deca19, 0, 1); local korb19 = If (stackb19 <=decb19, -1, 1);
local kora20 = If (stacka20 <=deca20, 0, 1); local korb20 = If (stackb20 <=decb20, -1, 1);
local kora21 = If (stacka21 <=deca21, 0, 1); local korb21 = If (stackb21 <=decb21, -1, 1);
local kora22 = If (stacka22 <=deca22, 0, 1); local korb22 = If (stackb22 <=decb22, -1, 1);
local kora23 = If (stacka23 <=deca23, 0, 1); local korb23 = If (stackb23 <=decb23, -1, 1);
local kora24 = If (stacka24 <=deca24, 0, 1); local korb24 = If (stackb24 <=decb24, -1, 1);
local kora25 = If (stacka25 <=deca25, 0, 1); local korb25 = If (stackb25 <=decb25, -1, 1);
local kora26 = If (stacka26 <=deca26, 0, 1); local korb26 = If (stackb26 <=decb26, -1, 1);
local kora27 = If (stacka27 <=deca27, 0, 1); local korb27 = If (stackb27 <=decb27, -1, 1);
local kora28 = If (stacka28 <=deca28, 0, 1); local korb28 = If (stackb28 <=decb28, -1, 1);
local kora29 = If (stacka29 <=deca29, 0, 1); local korb29 = If (stackb29 <=decb29, -1, 1);
local kora30 = If (stacka30 <=deca30, 0, 1); local korb30 = If (stackb30 <=decb30, -1, 1);
local kora31 = If (stacka31 <=deca31, 0, 1); local korb31 = If (stackb31 <=decb31, -1, 1);
local kora32 = If (stacka32 <=deca32, 0, 1); local korb32 = If (stackb32 <=decb32, -1, 1);
local kora33 = If (stacka33 <=deca33, 0, 1); local korb33 = If (stackb33 <=decb33, -1, 1);
local kora34 = If (stacka34 <=deca34, 0, 1); local korb34 = If (stackb34 <=decb34, -1, 1);
local kora35 = If (stacka35 <=deca35, 0, 1); local korb35 = If (stackb35 <=decb35, -1, 1);
local kora36 = If (stacka36 <=deca36, 0, 1); local korb36 = If (stackb36 <=decb36, -1, 1);

#####

'DEPENDENCE OF THE STACKING PROBABILITIES FROM THE PRECEDING LAYER

'parameters for (A γ B)(B α C)(C β A) stacking pattern

local ab2 = If (sx1 == 0, -1, 0); local bc2 = If (sx1 == -1/3, 1, 0); local ca2 = If (sx1 == 1/3, 1, 0);
local ab3 = If (sx2 == 0, -1, 0); local bc3 = If (sx2 == -1/3, 1, 0); local ca3 = If (sx2 == 1/3, 1, 0);
local ab4 = If (sx3 == 0, -1, 0); local bc4 = If (sx3 == -1/3, 1, 0); local ca4 = If (sx3 == 1/3, 1, 0);
local ab5 = If (sx4 == 0, -1, 0); local bc5 = If (sx4 == -1/3, 1, 0); local ca5 = If (sx4 == 1/3, 1, 0);
local ab6 = If (sx5 == 0, -1, 0); local bc6 = If (sx5 == -1/3, 1, 0); local ca6 = If (sx5 == 1/3, 1, 0);
local ab7 = If (sx6 == 0, -1, 0); local bc7 = If (sx6 == -1/3, 1, 0); local ca7 = If (sx6 == 1/3, 1, 0);
local ab8 = If (sx7 == 0, -1, 0); local bc8 = If (sx7 == -1/3, 1, 0); local ca8 = If (sx7 == 1/3, 1, 0);
local ab9 = If (sx8 == 0, -1, 0); local bc9 = If (sx8 == -1/3, 1, 0); local ca9 = If (sx8 == 1/3, 1, 0);
local ab10 = If (sx9 == 0, -1, 0); local bc10 = If (sx9 == -1/3, 1, 0); local ca10 = If (sx9 == 1/3, 1, 0);
local ab11 = If (sx10 == 0, -1, 0); local bc11 = If (sx10 == -1/3, 1, 0); local ca11 = If (sx10 == 1/3, 1, 0);
local ab12 = If (sx11 == 0, -1, 0); local bc12 = If (sx11 == -1/3, 1, 0); local ca12 = If (sx11 == 1/3, 1, 0);
local ab13 = If (sx12 == 0, -1, 0); local bc13 = If (sx12 == -1/3, 1, 0); local ca13 = If (sx12 == 1/3, 1, 0);
local ab14 = If (sx13 == 0, -1, 0); local bc14 = If (sx13 == -1/3, 1, 0); local ca14 = If (sx13 == 1/3, 1, 0);
local ab15 = If (sx14 == 0, -1, 0); local bc15 = If (sx14 == -1/3, 1, 0); local ca15 = If (sx14 == 1/3, 1, 0);


```

'parameters for (CβA)(BαC)(AγB) stacking pattern
local ac2 = If (sx1 == 0, 1, 0); local ba2 = If (sx1 == -1/3, 1, 0); local cb2 = If (sx1 == 1/3, -1, 0);
local ac3 = If (sx2 == 0, 1, 0); local ba3 = If (sx2 == -1/3, 1, 0); local cb3 = If (sx2 == 1/3, -1, 0);
local ac4 = If (sx3 == 0, 1, 0); local ba4 = If (sx3 == -1/3, 1, 0); local cb4 = If (sx3 == 1/3, -1, 0);
local ac5 = If (sx4 == 0, 1, 0); local ba5 = If (sx4 == -1/3, 1, 0); local cb5 = If (sx4 == 1/3, -1, 0);
local ac6 = If (sx5 == 0, 1, 0); local ba6 = If (sx5 == -1/3, 1, 0); local cb6 = If (sx5 == 1/3, -1, 0);
local ac7 = If (sx6 == 0, 1, 0); local ba7 = If (sx6 == -1/3, 1, 0); local cb7 = If (sx6 == 1/3, -1, 0);
local ac8 = If (sx7 == 0, 1, 0); local ba8 = If (sx7 == -1/3, 1, 0); local cb8 = If (sx7 == 1/3, -1, 0);
local ac9 = If (sx8 == 0, 1, 0); local ba9 = If (sx8 == -1/3, 1, 0); local cb9 = If (sx8 == 1/3, -1, 0);
local ac10 = If (sx9 == 0, 1, 0); local ba10 = If (sx9 == -1/3, 1, 0); local cb10 = If (sx9 == 1/3, -1, 0);
local ac11 = If (sx10 == 0, 1, 0); local ba11 = If (sx10 == -1/3, 1, 0); local cb11 = If (sx10 == 1/3, -1, 0);
local ac12 = If (sx11 == 0, 1, 0); local ba12 = If (sx11 == -1/3, 1, 0); local cb12 = If (sx11 == 1/3, -1, 0);
local ac13 = If (sx12 == 0, 1, 0); local ba13 = If (sx12 == -1/3, 1, 0); local cb13 = If (sx12 == 1/3, -1, 0);
local ac14 = If (sx13 == 0, 1, 0); local ba14 = If (sx13 == -1/3, 1, 0); local cb14 = If (sx13 == 1/3, -1, 0);
local ac15 = If (sx14 == 0, 1, 0); local ba15 = If (sx14 == -1/3, 1, 0); local cb15 = If (sx14 == 1/3, -1, 0);
local ac16 = If (sx15 == 0, 1, 0); local ba16 = If (sx15 == -1/3, 1, 0); local cb16 = If (sx15 == 1/3, -1, 0);
local ac17 = If (sx16 == 0, 1, 0); local ba17 = If (sx16 == -1/3, 1, 0); local cb17 = If (sx16 == 1/3, -1, 0);
local ac18 = If (sx17 == 0, 1, 0); local ba18 = If (sx17 == -1/3, 1, 0); local cb18 = If (sx17 == 1/3, -1, 0);
local ac19 = If (sx18 == 0, 1, 0); local ba19 = If (sx18 == -1/3, 1, 0); local cb19 = If (sx18 == 1/3, -1, 0);
local ac20 = If (sx19 == 0, 1, 0); local ba20 = If (sx19 == -1/3, 1, 0); local cb20 = If (sx19 == 1/3, -1, 0);
local ac21 = If (sx20 == 0, 1, 0); local ba21 = If (sx20 == -1/3, 1, 0); local cb21 = If (sx20 == 1/3, -1, 0);
local ac22 = If (sx21 == 0, 1, 0); local ba22 = If (sx21 == -1/3, 1, 0); local cb22 = If (sx21 == 1/3, -1, 0);
local ac23 = If (sx22 == 0, 1, 0); local ba23 = If (sx22 == -1/3, 1, 0); local cb23 = If (sx22 == 1/3, -1, 0);
local ac24 = If (sx23 == 0, 1, 0); local ba24 = If (sx23 == -1/3, 1, 0); local cb24 = If (sx23 == 1/3, -1, 0);
local ac25 = If (sx24 == 0, 1, 0); local ba25 = If (sx24 == -1/3, 1, 0); local cb25 = If (sx24 == 1/3, -1, 0);
local ac26 = If (sx25 == 0, 1, 0); local ba26 = If (sx25 == -1/3, 1, 0); local cb26 = If (sx25 == 1/3, -1, 0);
local ac27 = If (sx26 == 0, 1, 0); local ba27 = If (sx26 == -1/3, 1, 0); local cb27 = If (sx26 == 1/3, -1, 0);
local ac28 = If (sx27 == 0, 1, 0); local ba28 = If (sx27 == -1/3, 1, 0); local cb28 = If (sx27 == 1/3, -1, 0);
local ac29 = If (sx28 == 0, 1, 0); local ba29 = If (sx28 == -1/3, 1, 0); local cb29 = If (sx28 == 1/3, -1, 0);
local ac30 = If (sx29 == 0, 1, 0); local ba30 = If (sx29 == -1/3, 1, 0); local cb30 = If (sx29 == 1/3, -1, 0);
local ac31 = If (sx30 == 0, 1, 0); local ba31 = If (sx30 == -1/3, 1, 0); local cb31 = If (sx30 == 1/3, -1, 0);
local ac32 = If (sx31 == 0, 1, 0); local ba32 = If (sx31 == -1/3, 1, 0); local cb32 = If (sx31 == 1/3, -1, 0);
local ac33 = If (sx32 == 0, 1, 0); local ba33 = If (sx32 == -1/3, 1, 0); local cb33 = If (sx32 == 1/3, -1, 0);
local ac34 = If (sx33 == 0, 1, 0); local ba34 = If (sx33 == -1/3, 1, 0); local cb34 = If (sx33 == 1/3, -1, 0);
local ac35 = If (sx34 == 0, 1, 0); local ba35 = If (sx34 == -1/3, 1, 0); local cb35 = If (sx34 == 1/3, -1, 0);
local ac36 = If (sx35 == 0, 1, 0); local ba36 = If (sx35 == -1/3, 1, 0); local cb36 = If (sx35 == 1/3, -1, 0);
#####
#####
'SELECETION OF BASIS STACKING PATTERN

```

```

prn !sx0 = start/3;

```

```

'pabc-Parameter

```

```

local helper1pabc2 = If ((sx1 - sx0) == -1/3, 0, 1); local helper2pabc2 = If ((sx1 - sx0) == 2/3, 0, 1); local
helper3pabc2 = If ((helper2pabc2 * helper1pabc2) == 0, 1, 0);
local helper1pabc3 = If ((sx2 - sx1) == -1/3, 0, 1); local helper2pabc3 = If ((sx2 - sx1) == 2/3, 0, 1); local
helper3pabc3 = If ((helper2pabc3 * helper1pabc3) == 0, 1, 0);
local helper1pabc4 = If ((sx3 - sx2) == -1/3, 0, 1); local helper2pabc4 = If ((sx3 - sx2) == 2/3, 0, 1); local
helper3pabc4 = If ((helper2pabc4 * helper1pabc4) == 0, 1, 0);
local helper1pabc5 = If ((sx4 - sx3) == -1/3, 0, 1); local helper2pabc5 = If ((sx4 - sx3) == 2/3, 0, 1); local
helper3pabc5 = If ((helper2pabc5 * helper1pabc5) == 0, 1, 0);
local helper1pabc6 = If ((sx5 - sx4) == -1/3, 0, 1); local helper2pabc6 = If ((sx5 - sx4) == 2/3, 0, 1); local
helper3pabc6 = If ((helper2pabc6 * helper1pabc6) == 0, 1, 0);
local helper1pabc7 = If ((sx6 - sx5) == -1/3, 0, 1); local helper2pabc7 = If ((sx6 - sx5) == 2/3, 0, 1); local
helper3pabc7 = If ((helper2pabc7 * helper1pabc7) == 0, 1, 0);

```


local helper4pabc4 = If (helper1paa4 == 1, P21, P31);
 local helper4pabc5 = If (helper1paa5 == 1, P21, P31);
 local helper4pabc6 = If (helper1paa6 == 1, P21, P31);
 local helper4pabc7 = If (helper1paa7 == 1, P21, P31);
 local helper4pabc8 = If (helper1paa8 == 1, P21, P31);
 local helper4pabc9 = If (helper1paa9 == 1, P21, P31);
 local helper4pabc10 = If (helper1paa10 == 1, P21, P31);
 local helper4pabc11 = If (helper1paa11 == 1, P21, P31);
 local helper4pabc12 = If (helper1paa12 == 1, P21, P31);
 local helper4pabc13 = If (helper1paa13 == 1, P21, P31);
 local helper4pabc14 = If (helper1paa14 == 1, P21, P31);
 local helper4pabc15 = If (helper1paa15 == 1, P21, P31);
 local helper4pabc16 = If (helper1paa16 == 1, P21, P31);
 local helper4pabc17 = If (helper1paa17 == 1, P21, P31);
 local helper4pabc18 = If (helper1paa18 == 1, P21, P31);
 local helper4pabc19 = If (helper1paa19 == 1, P21, P31);
 local helper4pabc20 = If (helper1paa20 == 1, P21, P31);
 local helper4pabc21 = If (helper1paa21 == 1, P21, P31);
 local helper4pabc22 = If (helper1paa22 == 1, P21, P31);
 local helper4pabc23 = If (helper1paa23 == 1, P21, P31);
 local helper4pabc24 = If (helper1paa24 == 1, P21, P31);
 local helper4pabc25 = If (helper1paa25 == 1, P21, P31);
 local helper4pabc26 = If (helper1paa26 == 1, P21, P31);
 local helper4pabc27 = If (helper1paa27 == 1, P21, P31);
 local helper4pabc28 = If (helper1paa28 == 1, P21, P31);
 local helper4pabc29 = If (helper1paa29 == 1, P21, P31);
 local helper4pabc30 = If (helper1paa30 == 1, P21, P31);
 local helper4pabc31 = If (helper1paa31 == 1, P21, P31);
 local helper4pabc32 = If (helper1paa32 == 1, P21, P31);
 local helper4pabc33 = If (helper1paa33 == 1, P21, P31);
 local helper4pabc34 = If (helper1paa34 == 1, P21, P31);
 local helper4pabc35 = If (helper1paa35 == 1, P21, P31);
 local helper4pabc36 = If (helper1paa36 == 1, P21, P31);

local pabc2 = If (helper3pabc2 == 1, 1, helper4pabc2);
 local pabc3 = If (helper3pabc3 == 1, 1, helper4pabc3);
 local pabc4 = If (helper3pabc4 == 1, 1, helper4pabc4);
 local pabc5 = If (helper3pabc5 == 1, 1, helper4pabc5);
 local pabc6 = If (helper3pabc6 == 1, 1, helper4pabc6);
 local pabc7 = If (helper3pabc7 == 1, 1, helper4pabc7);
 local pabc8 = If (helper3pabc8 == 1, 1, helper4pabc8);
 local pabc9 = If (helper3pabc9 == 1, 1, helper4pabc9);
 local pabc10 = If (helper3pabc10 == 1, 1, helper4pabc10);
 local pabc11 = If (helper3pabc11 == 1, 1, helper4pabc11);
 local pabc12 = If (helper3pabc12 == 1, 1, helper4pabc12);
 local pabc13 = If (helper3pabc13 == 1, 1, helper4pabc13);
 local pabc14 = If (helper3pabc14 == 1, 1, helper4pabc14);
 local pabc15 = If (helper3pabc15 == 1, 1, helper4pabc15);
 local pabc16 = If (helper3pabc16 == 1, 1, helper4pabc16);
 local pabc17 = If (helper3pabc17 == 1, 1, helper4pabc17);
 local pabc18 = If (helper3pabc18 == 1, 1, helper4pabc18);
 local pabc19 = If (helper3pabc19 == 1, 1, helper4pabc19);
 local pabc20 = If (helper3pabc20 == 1, 1, helper4pabc20);
 local pabc21 = If (helper3pabc21 == 1, 1, helper4pabc21);
 local pabc22 = If (helper3pabc22 == 1, 1, helper4pabc22);
 local pabc23 = If (helper3pabc23 == 1, 1, helper4pabc23);
 local pabc24 = If (helper3pabc24 == 1, 1, helper4pabc24);
 local pabc25 = If (helper3pabc25 == 1, 1, helper4pabc25);
 local pabc26 = If (helper3pabc26 == 1, 1, helper4pabc26);
 local pabc27 = If (helper3pabc27 == 1, 1, helper4pabc27);
 local pabc28 = If (helper3pabc28 == 1, 1, helper4pabc28);

```

local pabc29 = If (helper3pabc29 == 1, 1, helper4pabc29);
local pabc30 = If (helper3pabc30 == 1, 1, helper4pabc30);
local pabc31 = If (helper3pabc31 == 1, 1, helper4pabc31);
local pabc32 = If (helper3pabc32 == 1, 1, helper4pabc32);
local pabc33 = If (helper3pabc33 == 1, 1, helper4pabc33);
local pabc34 = If (helper3pabc34 == 1, 1, helper4pabc34);
local pabc35 = If (helper3pabc35 == 1, 1, helper4pabc35);
local pabc36 = If (helper3pabc36 == 1, 1, helper4pabc36);

```

'paa-parameter

```

local helper1paa2 = If ((sx1 - sx0) == 0, 1, 0); local helper2paa2 = If (helper3pabc2 == 1, P12, P32);
local helper1paa3 = If ((sx2 - sx1) == 0, 1, 0); local helper2paa3 = If (helper3pabc3 == 1, P12, P32);
local helper1paa4 = If ((sx3 - sx2) == 0, 1, 0); local helper2paa4 = If (helper3pabc4 == 1, P12, P32);
local helper1paa5 = If ((sx4 - sx3) == 0, 1, 0); local helper2paa5 = If (helper3pabc5 == 1, P12, P32);
local helper1paa6 = If ((sx5 - sx4) == 0, 1, 0); local helper2paa6 = If (helper3pabc6 == 1, P12, P32);
local helper1paa7 = If ((sx6 - sx5) == 0, 1, 0); local helper2paa7 = If (helper3pabc7 == 1, P12, P32);
local helper1paa8 = If ((sx7 - sx6) == 0, 1, 0); local helper2paa8 = If (helper3pabc8 == 1, P12, P32);
local helper1paa9 = If ((sx8 - sx7) == 0, 1, 0); local helper2paa9 = If (helper3pabc9 == 1, P12, P32);
local helper1paa10 = If ((sx9 - sx8) == 0, 1, 0); local helper2paa10 = If (helper3pabc10 == 1, P12, P32);
local helper1paa11 = If ((sx10 - sx9) == 0, 1, 0); local helper2paa11 = If (helper3pabc11 == 1, P12, P32);
local helper1paa12 = If ((sx11 - sx10) == 0, 1, 0); local helper2paa12 = If (helper3pabc12 == 1, P12, P32);
local helper1paa13 = If ((sx12 - sx11) == 0, 1, 0); local helper2paa13 = If (helper3pabc13 == 1, P12, P32);
local helper1paa14 = If ((sx13 - sx12) == 0, 1, 0); local helper2paa14 = If (helper3pabc14 == 1, P12, P32);
local helper1paa15 = If ((sx14 - sx13) == 0, 1, 0); local helper2paa15 = If (helper3pabc15 == 1, P12, P32);
local helper1paa16 = If ((sx15 - sx14) == 0, 1, 0); local helper2paa16 = If (helper3pabc16 == 1, P12, P32);
local helper1paa17 = If ((sx16 - sx15) == 0, 1, 0); local helper2paa17 = If (helper3pabc17 == 1, P12, P32);
local helper1paa18 = If ((sx17 - sx16) == 0, 1, 0); local helper2paa18 = If (helper3pabc18 == 1, P12, P32);
local helper1paa19 = If ((sx18 - sx17) == 0, 1, 0); local helper2paa19 = If (helper3pabc19 == 1, P12, P32);
local helper1paa20 = If ((sx19 - sx18) == 0, 1, 0); local helper2paa20 = If (helper3pabc20 == 1, P12, P32);
local helper1paa21 = If ((sx20 - sx19) == 0, 1, 0); local helper2paa21 = If (helper3pabc21 == 1, P12, P32);
local helper1paa22 = If ((sx21 - sx20) == 0, 1, 0); local helper2paa22 = If (helper3pabc22 == 1, P12, P32);
local helper1paa23 = If ((sx22 - sx21) == 0, 1, 0); local helper2paa23 = If (helper3pabc23 == 1, P12, P32);
local helper1paa24 = If ((sx23 - sx22) == 0, 1, 0); local helper2paa24 = If (helper3pabc24 == 1, P12, P32);
local helper1paa25 = If ((sx24 - sx23) == 0, 1, 0); local helper2paa25 = If (helper3pabc25 == 1, P12, P32);
local helper1paa26 = If ((sx25 - sx24) == 0, 1, 0); local helper2paa26 = If (helper3pabc26 == 1, P12, P32);
local helper1paa27 = If ((sx26 - sx25) == 0, 1, 0); local helper2paa27 = If (helper3pabc27 == 1, P12, P32);
local helper1paa28 = If ((sx27 - sx26) == 0, 1, 0); local helper2paa28 = If (helper3pabc28 == 1, P12, P32);
local helper1paa29 = If ((sx28 - sx27) == 0, 1, 0); local helper2paa29 = If (helper3pabc29 == 1, P12, P32);
local helper1paa30 = If ((sx29 - sx28) == 0, 1, 0); local helper2paa30 = If (helper3pabc30 == 1, P12, P32);
local helper1paa31 = If ((sx30 - sx29) == 0, 1, 0); local helper2paa31 = If (helper3pabc31 == 1, P12, P32);
local helper1paa32 = If ((sx31 - sx30) == 0, 1, 0); local helper2paa32 = If (helper3pabc32 == 1, P12, P32);
local helper1paa33 = If ((sx32 - sx31) == 0, 1, 0); local helper2paa33 = If (helper3pabc33 == 1, P12, P32);
local helper1paa34 = If ((sx33 - sx32) == 0, 1, 0); local helper2paa34 = If (helper3pabc34 == 1, P12, P32);
local helper1paa35 = If ((sx34 - sx33) == 0, 1, 0); local helper2paa35 = If (helper3pabc35 == 1, P12, P32);
local helper1paa36 = If ((sx35 - sx34) == 0, 1, 0); local helper2paa36 = If (helper3pabc36 == 1, P12, P32);

```

```

local paa2 = If (helper1paa2 == 1, 1, helper2paa2 );
local paa3 = If (helper1paa3 == 1, 1, helper2paa3 );
local paa4 = If (helper1paa4 == 1, 1, helper2paa4 );
local paa5 = If (helper1paa5 == 1, 1, helper2paa5 );
local paa6 = If (helper1paa6 == 1, 1, helper2paa6 );
local paa7 = If (helper1paa7 == 1, 1, helper2paa7 );
local paa8 = If (helper1paa8 == 1, 1, helper2paa8 );
local paa9 = If (helper1paa9 == 1, 1, helper2paa9 );
local paa10 = If (helper1paa10 == 1, 1, helper2paa10);
local paa11 = If (helper1paa11 == 1, 1, helper2paa11);
local paa12 = If (helper1paa12 == 1, 1, helper2paa12);
local paa13 = If (helper1paa13 == 1, 1, helper2paa13);
local paa14 = If (helper1paa14 == 1, 1, helper2paa14);
local paa15 = If (helper1paa15 == 1, 1, helper2paa15);

```

```

local paa16 = If (helper1paa16 == 1, 1, helper2paa16);
local paa17 = If (helper1paa17 == 1, 1, helper2paa17);
local paa18 = If (helper1paa18 == 1, 1, helper2paa18);
local paa19 = If (helper1paa19 == 1, 1, helper2paa19);
local paa20 = If (helper1paa20 == 1, 1, helper2paa20);
local paa21 = If (helper1paa21 == 1, 1, helper2paa21);
local paa22 = If (helper1paa22 == 1, 1, helper2paa22);
local paa23 = If (helper1paa23 == 1, 1, helper2paa23);
local paa24 = If (helper1paa24 == 1, 1, helper2paa24);
local paa25 = If (helper1paa25 == 1, 1, helper2paa25);
local paa26 = If (helper1paa26 == 1, 1, helper2paa26);
local paa27 = If (helper1paa27 == 1, 1, helper2paa27);
local paa28 = If (helper1paa28 == 1, 1, helper2paa28);
local paa29 = If (helper1paa29 == 1, 1, helper2paa29);
local paa30 = If (helper1paa30 == 1, 1, helper2paa30);
local paa31 = If (helper1paa31 == 1, 1, helper2paa31);
local paa32 = If (helper1paa32 == 1, 1, helper2paa32);
local paa33 = If (helper1paa33 == 1, 1, helper2paa33);
local paa34 = If (helper1paa34 == 1, 1, helper2paa34);
local paa35 = If (helper1paa35 == 1, 1, helper2paa35);
local paa36 = If (helper1paa36 == 1, 1, helper2paa36);

```

pcba-Parameter

```

local helper1pab2 = If ((sx1 - sx0) == 1/3, 0, 1); local helper2pab2 = If ((sx1 - sx0) == -2/3, 0, 1); local
helper3pab2 = If ((helper2pab2 * helper1pab2) == 0, 1, 0);
local helper1pab3 = If ((sx2 - sx1) == 1/3, 0, 1); local helper2pab3 = If ((sx2 - sx1) == -2/3, 0, 1); local
helper3pab3 = If ((helper2pab3 * helper1pab3) == 0, 1, 0);
local helper1pab4 = If ((sx3 - sx2) == 1/3, 0, 1); local helper2pab4 = If ((sx3 - sx2) == -2/3, 0, 1); local
helper3pab4 = If ((helper2pab4 * helper1pab4) == 0, 1, 0);
local helper1pab5 = If ((sx4 - sx3) == 1/3, 0, 1); local helper2pab5 = If ((sx4 - sx3) == -2/3, 0, 1); local
helper3pab5 = If ((helper2pab5 * helper1pab5) == 0, 1, 0);
local helper1pab6 = If ((sx5 - sx4) == 1/3, 0, 1); local helper2pab6 = If ((sx5 - sx4) == -2/3, 0, 1); local
helper3pab6 = If ((helper2pab6 * helper1pab6) == 0, 1, 0);
local helper1pab7 = If ((sx6 - sx5) == 1/3, 0, 1); local helper2pab7 = If ((sx6 - sx5) == -2/3, 0, 1); local
helper3pab7 = If ((helper2pab7 * helper1pab7) == 0, 1, 0);
local helper1pab8 = If ((sx7 - sx6) == 1/3, 0, 1); local helper2pab8 = If ((sx7 - sx6) == -2/3, 0, 1); local
helper3pab8 = If ((helper2pab8 * helper1pab8) == 0, 1, 0);
local helper1pab9 = If ((sx8 - sx7) == 1/3, 0, 1); local helper2pab9 = If ((sx8 - sx7) == -2/3, 0, 1); local
helper3pab9 = If ((helper2pab9 * helper1pab9) == 0, 1, 0);
local helper1pab10 = If ((sx9 - sx8) == 1/3, 0, 1); local helper2pab10 = If ((sx9 - sx8) == -2/3, 0, 1); local
helper3pab10 = If ((helper2pab10 * helper1pab10) == 0, 1, 0);
local helper1pab11 = If ((sx10 - sx9) == 1/3, 0, 1); local helper2pab11 = If ((sx10 - sx9) == -2/3, 0, 1); local
helper3pab11 = If ((helper2pab11 * helper1pab11) == 0, 1, 0);
local helper1pab12 = If ((sx11 - sx10) == 1/3, 0, 1); local helper2pab12 = If ((sx11 - sx10) == -2/3, 0, 1); local
helper3pab12 = If ((helper2pab12 * helper1pab12) == 0, 1, 0);
local helper1pab13 = If ((sx12 - sx11) == 1/3, 0, 1); local helper2pab13 = If ((sx12 - sx11) == -2/3, 0, 1); local
helper3pab13 = If ((helper2pab13 * helper1pab13) == 0, 1, 0);
local helper1pab14 = If ((sx13 - sx12) == 1/3, 0, 1); local helper2pab14 = If ((sx13 - sx12) == -2/3, 0, 1); local
helper3pab14 = If ((helper2pab14 * helper1pab14) == 0, 1, 0);
local helper1pab15 = If ((sx14 - sx13) == 1/3, 0, 1); local helper2pab15 = If ((sx14 - sx13) == -2/3, 0, 1); local
helper3pab15 = If ((helper2pab15 * helper1pab15) == 0, 1, 0);
local helper1pab16 = If ((sx15 - sx14) == 1/3, 0, 1); local helper2pab16 = If ((sx15 - sx14) == -2/3, 0, 1); local
helper3pab16 = If ((helper2pab16 * helper1pab16) == 0, 1, 0);
local helper1pab17 = If ((sx16 - sx15) == 1/3, 0, 1); local helper2pab17 = If ((sx16 - sx15) == -2/3, 0, 1); local
helper3pab17 = If ((helper2pab17 * helper1pab17) == 0, 1, 0);
local helper1pab18 = If ((sx17 - sx16) == 1/3, 0, 1); local helper2pab18 = If ((sx17 - sx16) == -2/3, 0, 1); local
helper3pab18 = If ((helper2pab18 * helper1pab18) == 0, 1, 0);
local helper1pab19 = If ((sx18 - sx17) == 1/3, 0, 1); local helper2pab19 = If ((sx18 - sx17) == -2/3, 0, 1); local
helper3pab19 = If ((helper2pab19 * helper1pab19) == 0, 1, 0);

```

local helper1pab20 = If ((sx19 - sx18) == 1/3, 0, 1); local helper2pab20 = If ((sx19 - sx18) == -2/3, 0, 1); local
 helper3pab20 = If ((helper2pab20 * helper1pab20) == 0, 1, 0);
 local helper1pab21 = If ((sx20 - sx19) == 1/3, 0, 1); local helper2pab21 = If ((sx20 - sx19) == -2/3, 0, 1); local
 helper3pab21 = If ((helper2pab21 * helper1pab21) == 0, 1, 0);
 local helper1pab22 = If ((sx21 - sx20) == 1/3, 0, 1); local helper2pab22 = If ((sx21 - sx20) == -2/3, 0, 1); local
 helper3pab22 = If ((helper2pab22 * helper1pab22) == 0, 1, 0);
 local helper1pab23 = If ((sx22 - sx21) == 1/3, 0, 1); local helper2pab23 = If ((sx22 - sx21) == -2/3, 0, 1); local
 helper3pab23 = If ((helper2pab23 * helper1pab23) == 0, 1, 0);
 local helper1pab24 = If ((sx23 - sx22) == 1/3, 0, 1); local helper2pab24 = If ((sx23 - sx22) == -2/3, 0, 1); local
 helper3pab24 = If ((helper2pab24 * helper1pab24) == 0, 1, 0);
 local helper1pab25 = If ((sx24 - sx23) == 1/3, 0, 1); local helper2pab25 = If ((sx24 - sx23) == -2/3, 0, 1); local
 helper3pab25 = If ((helper2pab25 * helper1pab25) == 0, 1, 0);
 local helper1pab26 = If ((sx25 - sx24) == 1/3, 0, 1); local helper2pab26 = If ((sx25 - sx24) == -2/3, 0, 1); local
 helper3pab26 = If ((helper2pab26 * helper1pab26) == 0, 1, 0);
 local helper1pab27 = If ((sx26 - sx25) == 1/3, 0, 1); local helper2pab27 = If ((sx26 - sx25) == -2/3, 0, 1); local
 helper3pab27 = If ((helper2pab27 * helper1pab27) == 0, 1, 0);
 local helper1pab28 = If ((sx27 - sx26) == 1/3, 0, 1); local helper2pab28 = If ((sx27 - sx26) == -2/3, 0, 1); local
 helper3pab28 = If ((helper2pab28 * helper1pab28) == 0, 1, 0);
 local helper1pab29 = If ((sx28 - sx27) == 1/3, 0, 1); local helper2pab29 = If ((sx28 - sx27) == -2/3, 0, 1); local
 helper3pab29 = If ((helper2pab29 * helper1pab29) == 0, 1, 0);
 local helper1pab30 = If ((sx29 - sx28) == 1/3, 0, 1); local helper2pab30 = If ((sx29 - sx28) == -2/3, 0, 1); local
 helper3pab30 = If ((helper2pab30 * helper1pab30) == 0, 1, 0);
 local helper1pab31 = If ((sx30 - sx29) == 1/3, 0, 1); local helper2pab31 = If ((sx30 - sx29) == -2/3, 0, 1); local
 helper3pab31 = If ((helper2pab31 * helper1pab31) == 0, 1, 0);
 local helper1pab32 = If ((sx31 - sx30) == 1/3, 0, 1); local helper2pab32 = If ((sx31 - sx30) == -2/3, 0, 1); local
 helper3pab32 = If ((helper2pab32 * helper1pab32) == 0, 1, 0);
 local helper1pab33 = If ((sx32 - sx31) == 1/3, 0, 1); local helper2pab33 = If ((sx32 - sx31) == -2/3, 0, 1); local
 helper3pab33 = If ((helper2pab33 * helper1pab33) == 0, 1, 0);
 local helper1pab34 = If ((sx33 - sx32) == 1/3, 0, 1); local helper2pab34 = If ((sx33 - sx32) == -2/3, 0, 1); local
 helper3pab34 = If ((helper2pab34 * helper1pab34) == 0, 1, 0);
 local helper1pab35 = If ((sx34 - sx33) == 1/3, 0, 1); local helper2pab35 = If ((sx34 - sx33) == -2/3, 0, 1); local
 helper3pab35 = If ((helper2pab35 * helper1pab35) == 0, 1, 0);
 local helper1pab36 = If ((sx35 - sx34) == 1/3, 0, 1); local helper2pab36 = If ((sx35 - sx34) == -2/3, 0, 1); local
 helper3pab36 = If ((helper2pab36 * helper1pab36) == 0, 1, 0);

local helper4pab2 = If (helper1paa2 == 1, P23, P13);
 local helper4pab3 = If (helper1paa3 == 1, P23, P13);
 local helper4pab4 = If (helper1paa4 == 1, P23, P13);
 local helper4pab5 = If (helper1paa5 == 1, P23, P13);
 local helper4pab6 = If (helper1paa6 == 1, P23, P13);
 local helper4pab7 = If (helper1paa7 == 1, P23, P13);
 local helper4pab8 = If (helper1paa8 == 1, P23, P13);
 local helper4pab9 = If (helper1paa9 == 1, P23, P13);
 local helper4pab10 = If (helper1paa10 == 1, P23, P13);
 local helper4pab11 = If (helper1paa11 == 1, P23, P13);
 local helper4pab12 = If (helper1paa12 == 1, P23, P13);
 local helper4pab13 = If (helper1paa13 == 1, P23, P13);
 local helper4pab14 = If (helper1paa14 == 1, P23, P13);
 local helper4pab15 = If (helper1paa15 == 1, P23, P13);
 local helper4pab16 = If (helper1paa16 == 1, P23, P13);
 local helper4pab17 = If (helper1paa17 == 1, P23, P13);
 local helper4pab18 = If (helper1paa18 == 1, P23, P13);
 local helper4pab19 = If (helper1paa19 == 1, P23, P13);
 local helper4pab20 = If (helper1paa20 == 1, P23, P13);
 local helper4pab21 = If (helper1paa21 == 1, P23, P13);
 local helper4pab22 = If (helper1paa22 == 1, P23, P13);
 local helper4pab23 = If (helper1paa23 == 1, P23, P13);
 local helper4pab24 = If (helper1paa24 == 1, P23, P13);
 local helper4pab25 = If (helper1paa25 == 1, P23, P13);
 local helper4pab26 = If (helper1paa26 == 1, P23, P13);
 local helper4pab27 = If (helper1paa27 == 1, P23, P13);

```
local helper4pab28 = If (helper1paa28 == 1, P23, P13);
local helper4pab29 = If (helper1paa29 == 1, P23, P13);
local helper4pab30 = If (helper1paa30 == 1, P23, P13);
local helper4pab31 = If (helper1paa31 == 1, P23, P13);
local helper4pab32 = If (helper1paa32 == 1, P23, P13);
local helper4pab33 = If (helper1paa33 == 1, P23, P13);
local helper4pab34 = If (helper1paa34 == 1, P23, P13);
local helper4pab35 = If (helper1paa35 == 1, P23, P13);
local helper4pab36 = If (helper1paa36 == 1, P23, P13);
```

```
local pab2 = If (helper3pab2 == 1, 1, helper4pab2 );
local pab3 = If (helper3pab3 == 1, 1, helper4pab3 );
local pab4 = If (helper3pab4 == 1, 1, helper4pab4 );
local pab5 = If (helper3pab5 == 1, 1, helper4pab5 );
local pab6 = If (helper3pab6 == 1, 1, helper4pab6 );
local pab7 = If (helper3pab7 == 1, 1, helper4pab7 );
local pab8 = If (helper3pab8 == 1, 1, helper4pab8 );
local pab9 = If (helper3pab9 == 1, 1, helper4pab9 );
local pab10 = If (helper3pab10 == 1, 1, helper4pab10);
local pab11 = If (helper3pab11 == 1, 1, helper4pab11);
local pab12 = If (helper3pab12 == 1, 1, helper4pab12);
local pab13 = If (helper3pab13 == 1, 1, helper4pab13);
local pab14 = If (helper3pab14 == 1, 1, helper4pab14);
local pab15 = If (helper3pab15 == 1, 1, helper4pab15);
local pab16 = If (helper3pab16 == 1, 1, helper4pab16);
local pab17 = If (helper3pab17 == 1, 1, helper4pab17);
local pab18 = If (helper3pab18 == 1, 1, helper4pab18);
local pab19 = If (helper3pab19 == 1, 1, helper4pab19);
local pab20 = If (helper3pab20 == 1, 1, helper4pab20);
local pab21 = If (helper3pab21 == 1, 1, helper4pab21);
local pab22 = If (helper3pab22 == 1, 1, helper4pab22);
local pab23 = If (helper3pab23 == 1, 1, helper4pab23);
local pab24 = If (helper3pab24 == 1, 1, helper4pab24);
local pab25 = If (helper3pab25 == 1, 1, helper4pab25);
local pab26 = If (helper3pab26 == 1, 1, helper4pab26);
local pab27 = If (helper3pab27 == 1, 1, helper4pab27);
local pab28 = If (helper3pab28 == 1, 1, helper4pab28);
local pab29 = If (helper3pab29 == 1, 1, helper4pab29);
local pab30 = If (helper3pab30 == 1, 1, helper4pab30);
local pab31 = If (helper3pab31 == 1, 1, helper4pab31);
local pab32 = If (helper3pab32 == 1, 1, helper4pab32);
local pab33 = If (helper3pab33 == 1, 1, helper4pab33);
local pab34 = If (helper3pab34 == 1, 1, helper4pab34);
local pab35 = If (helper3pab35 == 1, 1, helper4pab35);
local pab36 = If (helper3pab36 == 1, 1, helper4pab36);
```

```
#####
#####
```

'SWITCHES FOR SCALING FACTORS

```
local sabc2 = If (pabc2 == 1, 1, 0); local saa2 = If (paa2 == 1, 1, 0); local sab2 = If (pab2 == 1, 1, 0);
local sabc3 = If (pabc3 == 1, 1, 0); local saa3 = If (paa3 == 1, 1, 0); local sab3 = If (pab3 == 1, 1, 0);
local sabc4 = If (pabc4 == 1, 1, 0); local saa4 = If (paa4 == 1, 1, 0); local sab4 = If (pab4 == 1, 1, 0);
local sabc5 = If (pabc5 == 1, 1, 0); local saa5 = If (paa5 == 1, 1, 0); local sab5 = If (pab5 == 1, 1, 0);
local sabc6 = If (pabc6 == 1, 1, 0); local saa6 = If (paa6 == 1, 1, 0); local sab6 = If (pab6 == 1, 1, 0);
local sabc7 = If (pabc7 == 1, 1, 0); local saa7 = If (paa7 == 1, 1, 0); local sab7 = If (pab7 == 1, 1, 0);
local sabc8 = If (pabc8 == 1, 1, 0); local saa8 = If (paa8 == 1, 1, 0); local sab8 = If (pab8 == 1, 1, 0);
local sabc9 = If (pabc9 == 1, 1, 0); local saa9 = If (paa9 == 1, 1, 0); local sab9 = If (pab9 == 1, 1, 0);
local sabc10 = If (pabc10 == 1, 1, 0); local saa10 = If (paa10 == 1, 1, 0); local sab10 = If (pab10 == 1, 1, 0);
local sabc11 = If (pabc11 == 1, 1, 0); local saa11 = If (paa11 == 1, 1, 0); local sab11 = If (pab11 == 1, 1, 0);
local sabc12 = If (pabc12 == 1, 1, 0); local saa12 = If (paa12 == 1, 1, 0); local sab12 = If (pab12 == 1, 1, 0);
local sabc13 = If (pabc13 == 1, 1, 0); local saa13 = If (paa13 == 1, 1, 0); local sab13 = If (pab13 == 1, 1, 0);
local sabc14 = If (pabc14 == 1, 1, 0); local saa14 = If (paa14 == 1, 1, 0); local sab14 = If (pab14 == 1, 1, 0);
```

```

local sabc15 = If (pabc15 == 1, 1, 0); local saa15 = If (paa15 == 1, 1, 0); local sab15 = If (pab15 == 1, 1, 0);
local sabc16 = If (pabc16 == 1, 1, 0); local saa16 = If (paa16 == 1, 1, 0); local sab16 = If (pab16 == 1, 1, 0);
local sabc17 = If (pabc17 == 1, 1, 0); local saa17 = If (paa17 == 1, 1, 0); local sab17 = If (pab17 == 1, 1, 0);
local sabc18 = If (pabc18 == 1, 1, 0); local saa18 = If (paa18 == 1, 1, 0); local sab18 = If (pab18 == 1, 1, 0);
local sabc19 = If (pabc19 == 1, 1, 0); local saa19 = If (paa19 == 1, 1, 0); local sab19 = If (pab19 == 1, 1, 0);
local sabc20 = If (pabc20 == 1, 1, 0); local saa20 = If (paa20 == 1, 1, 0); local sab20 = If (pab20 == 1, 1, 0);
local sabc21 = If (pabc21 == 1, 1, 0); local saa21 = If (paa21 == 1, 1, 0); local sab21 = If (pab21 == 1, 1, 0);
local sabc22 = If (pabc22 == 1, 1, 0); local saa22 = If (paa22 == 1, 1, 0); local sab22 = If (pab22 == 1, 1, 0);
local sabc23 = If (pabc23 == 1, 1, 0); local saa23 = If (paa23 == 1, 1, 0); local sab23 = If (pab23 == 1, 1, 0);
local sabc24 = If (pabc24 == 1, 1, 0); local saa24 = If (paa24 == 1, 1, 0); local sab24 = If (pab24 == 1, 1, 0);
local sabc25 = If (pabc25 == 1, 1, 0); local saa25 = If (paa25 == 1, 1, 0); local sab25 = If (pab25 == 1, 1, 0);
local sabc26 = If (pabc26 == 1, 1, 0); local saa26 = If (paa26 == 1, 1, 0); local sab26 = If (pab26 == 1, 1, 0);
local sabc27 = If (pabc27 == 1, 1, 0); local saa27 = If (paa27 == 1, 1, 0); local sab27 = If (pab27 == 1, 1, 0);
local sabc28 = If (pabc28 == 1, 1, 0); local saa28 = If (paa28 == 1, 1, 0); local sab28 = If (pab28 == 1, 1, 0);
local sabc29 = If (pabc29 == 1, 1, 0); local saa29 = If (paa29 == 1, 1, 0); local sab29 = If (pab29 == 1, 1, 0);
local sabc30 = If (pabc30 == 1, 1, 0); local saa30 = If (paa30 == 1, 1, 0); local sab30 = If (pab30 == 1, 1, 0);
local sabc31 = If (pabc31 == 1, 1, 0); local saa31 = If (paa31 == 1, 1, 0); local sab31 = If (pab31 == 1, 1, 0);
local sabc32 = If (pabc32 == 1, 1, 0); local saa32 = If (paa32 == 1, 1, 0); local sab32 = If (pab32 == 1, 1, 0);
local sabc33 = If (pabc33 == 1, 1, 0); local saa33 = If (paa33 == 1, 1, 0); local sab33 = If (pab33 == 1, 1, 0);
local sabc34 = If (pabc34 == 1, 1, 0); local saa34 = If (paa34 == 1, 1, 0); local sab34 = If (pab34 == 1, 1, 0);
local sabc35 = If (pabc35 == 1, 1, 0); local saa35 = If (paa35 == 1, 1, 0); local sab35 = If (pab35 == 1, 1, 0);
local sabc36 = If (pabc36 == 1, 1, 0); local saa36 = If (paa36 == 1, 1, 0); local sab36 = If (pab36 == 1, 1, 0);

```

```

-----
local saaab2 = If (paa2 + pab2 == 0, 1, 0);
local saaab3 = If (paa3 + pab3 == 0, 1, 0);
local saaab4 = If (paa4 + pab4 == 0, 1, 0);
local saaab5 = If (paa5 + pab5 == 0, 1, 0);
local saaab6 = If (paa6 + pab6 == 0, 1, 0);
local saaab7 = If (paa7 + pab7 == 0, 1, 0);
local saaab8 = If (paa8 + pab8 == 0, 1, 0);
local saaab9 = If (paa9 + pab9 == 0, 1, 0);
local saaab10 = If (paa10 + pab10 == 0, 1, 0);
local saaab11 = If (paa11 + pab11 == 0, 1, 0);
local saaab12 = If (paa12 + pab12 == 0, 1, 0);
local saaab13 = If (paa13 + pab13 == 0, 1, 0);
local saaab14 = If (paa14 + pab14 == 0, 1, 0);
local saaab15 = If (paa15 + pab15 == 0, 1, 0);
local saaab16 = If (paa16 + pab16 == 0, 1, 0);
local saaab17 = If (paa17 + pab17 == 0, 1, 0);
local saaab18 = If (paa18 + pab18 == 0, 1, 0);
local saaab19 = If (paa19 + pab19 == 0, 1, 0);
local saaab20 = If (paa20 + pab20 == 0, 1, 0);
local saaab21 = If (paa21 + pab21 == 0, 1, 0);
local saaab22 = If (paa22 + pab22 == 0, 1, 0);
local saaab23 = If (paa23 + pab23 == 0, 1, 0);
local saaab24 = If (paa24 + pab24 == 0, 1, 0);
local saaab25 = If (paa25 + pab25 == 0, 1, 0);
local saaab26 = If (paa26 + pab26 == 0, 1, 0);
local saaab27 = If (paa27 + pab27 == 0, 1, 0);
local saaab28 = If (paa28 + pab28 == 0, 1, 0);
local saaab29 = If (paa29 + pab29 == 0, 1, 0);
local saaab30 = If (paa30 + pab30 == 0, 1, 0);
local saaab31 = If (paa31 + pab31 == 0, 1, 0);
local saaab32 = If (paa32 + pab32 == 0, 1, 0);
local saaab33 = If (paa33 + pab33 == 0, 1, 0);
local saaab34 = If (paa34 + pab34 == 0, 1, 0);
local saaab35 = If (paa35 + pab35 == 0, 1, 0);
local saaab36 = If (paa36 + pab36 == 0, 1, 0);

```



```

local sabcab2 = If (pabc2 + pab2 == 0, 1, 0); local sabcaa2 = If (pabc2 + paa2 == 0, 1, 0);
local sabcab3 = If (pabc3 + pab3 == 0, 1, 0); local sabcaa3 = If (pabc3 + paa3 == 0, 1, 0);
local sabcab4 = If (pabc4 + pab4 == 0, 1, 0); local sabcaa4 = If (pabc4 + paa4 == 0, 1, 0);
local sabcab5 = If (pabc5 + pab5 == 0, 1, 0); local sabcaa5 = If (pabc5 + paa5 == 0, 1, 0);
local sabcab6 = If (pabc6 + pab6 == 0, 1, 0); local sabcaa6 = If (pabc6 + paa6 == 0, 1, 0);
local sabcab7 = If (pabc7 + pab7 == 0, 1, 0); local sabcaa7 = If (pabc7 + paa7 == 0, 1, 0);
local sabcab8 = If (pabc8 + pab8 == 0, 1, 0); local sabcaa8 = If (pabc8 + paa8 == 0, 1, 0);
local sabcab9 = If (pabc9 + pab9 == 0, 1, 0); local sabcaa9 = If (pabc9 + paa9 == 0, 1, 0);
local sabcab10 = If (pabc10 + pab10 == 0, 1, 0); local sabcaa10 = If (pabc10 + paa10 == 0, 1, 0);
local sabcab11 = If (pabc11 + pab11 == 0, 1, 0); local sabcaa11 = If (pabc11 + paa11 == 0, 1, 0);
local sabcab12 = If (pabc12 + pab12 == 0, 1, 0); local sabcaa12 = If (pabc12 + paa12 == 0, 1, 0);
local sabcab13 = If (pabc13 + pab13 == 0, 1, 0); local sabcaa13 = If (pabc13 + paa13 == 0, 1, 0);
local sabcab14 = If (pabc14 + pab14 == 0, 1, 0); local sabcaa14 = If (pabc14 + paa14 == 0, 1, 0);
local sabcab15 = If (pabc15 + pab15 == 0, 1, 0); local sabcaa15 = If (pabc15 + paa15 == 0, 1, 0);
local sabcab16 = If (pabc16 + pab16 == 0, 1, 0); local sabcaa16 = If (pabc16 + paa16 == 0, 1, 0);
local sabcab17 = If (pabc17 + pab17 == 0, 1, 0); local sabcaa17 = If (pabc17 + paa17 == 0, 1, 0);
local sabcab18 = If (pabc18 + pab18 == 0, 1, 0); local sabcaa18 = If (pabc18 + paa18 == 0, 1, 0);
local sabcab19 = If (pabc19 + pab19 == 0, 1, 0); local sabcaa19 = If (pabc19 + paa19 == 0, 1, 0);
local sabcab20 = If (pabc20 + pab20 == 0, 1, 0); local sabcaa20 = If (pabc20 + paa20 == 0, 1, 0);
local sabcab21 = If (pabc21 + pab21 == 0, 1, 0); local sabcaa21 = If (pabc21 + paa21 == 0, 1, 0);
local sabcab22 = If (pabc22 + pab22 == 0, 1, 0); local sabcaa22 = If (pabc22 + paa22 == 0, 1, 0);
local sabcab23 = If (pabc23 + pab23 == 0, 1, 0); local sabcaa23 = If (pabc23 + paa23 == 0, 1, 0);
local sabcab24 = If (pabc24 + pab24 == 0, 1, 0); local sabcaa24 = If (pabc24 + paa24 == 0, 1, 0);
local sabcab25 = If (pabc25 + pab25 == 0, 1, 0); local sabcaa25 = If (pabc25 + paa25 == 0, 1, 0);
local sabcab26 = If (pabc26 + pab26 == 0, 1, 0); local sabcaa26 = If (pabc26 + paa26 == 0, 1, 0);
local sabcab27 = If (pabc27 + pab27 == 0, 1, 0); local sabcaa27 = If (pabc27 + paa27 == 0, 1, 0);
local sabcab28 = If (pabc28 + pab28 == 0, 1, 0); local sabcaa28 = If (pabc28 + paa28 == 0, 1, 0);
local sabcab29 = If (pabc29 + pab29 == 0, 1, 0); local sabcaa29 = If (pabc29 + paa29 == 0, 1, 0);
local sabcab30 = If (pabc30 + pab30 == 0, 1, 0); local sabcaa30 = If (pabc30 + paa30 == 0, 1, 0);
local sabcab31 = If (pabc31 + pab31 == 0, 1, 0); local sabcaa31 = If (pabc31 + paa31 == 0, 1, 0);
local sabcab32 = If (pabc32 + pab32 == 0, 1, 0); local sabcaa32 = If (pabc32 + paa32 == 0, 1, 0);
local sabcab33 = If (pabc33 + pab33 == 0, 1, 0); local sabcaa33 = If (pabc33 + paa33 == 0, 1, 0);
local sabcab34 = If (pabc34 + pab34 == 0, 1, 0); local sabcaa34 = If (pabc34 + paa34 == 0, 1, 0);
local sabcab35 = If (pabc35 + pab35 == 0, 1, 0); local sabcaa35 = If (pabc35 + paa35 == 0, 1, 0);
local sabcab36 = If (pabc36 + pab36 == 0, 1, 0); local sabcaa36 = If (pabc36 + paa36 == 0, 1, 0);
'#####
#####

```

'SWITCHES FOR BASIS STACKING PATTERN

```

local ssabc2 = If (pabc2 == 1, 1, 0); local ssa2 = If (paa2 == 1, 1, 0); local ssab2 = If (pab2 == 1, 1, 0);
local ssabc3 = If (pabc3 == 1, 1, 0); local ssa3 = If (paa3 == 1, 1, 0); local ssab3 = If (pab3 == 1, 1, 0);
local ssabc4 = If (pabc4 == 1, 1, 0); local ssa4 = If (paa4 == 1, 1, 0); local ssab4 = If (pab4 == 1, 1, 0);
local ssabc5 = If (pabc5 == 1, 1, 0); local ssa5 = If (paa5 == 1, 1, 0); local ssab5 = If (pab5 == 1, 1, 0);
local ssabc6 = If (pabc6 == 1, 1, 0); local ssa6 = If (paa6 == 1, 1, 0); local ssab6 = If (pab6 == 1, 1, 0);
local ssabc7 = If (pabc7 == 1, 1, 0); local ssa7 = If (paa7 == 1, 1, 0); local ssab7 = If (pab7 == 1, 1, 0);
local ssabc8 = If (pabc8 == 1, 1, 0); local ssa8 = If (paa8 == 1, 1, 0); local ssab8 = If (pab8 == 1, 1, 0);
local ssabc9 = If (pabc9 == 1, 1, 0); local ssa9 = If (paa9 == 1, 1, 0); local ssab9 = If (pab9 == 1, 1, 0);
local ssabc10 = If (pabc10 == 1, 1, 0); local ssa10 = If (paa10 == 1, 1, 0); local ssab10 = If (pab10 == 1, 1, 0);
local ssabc11 = If (pabc11 == 1, 1, 0); local ssa11 = If (paa11 == 1, 1, 0); local ssab11 = If (pab11 == 1, 1, 0);
local ssabc12 = If (pabc12 == 1, 1, 0); local ssa12 = If (paa12 == 1, 1, 0); local ssab12 = If (pab12 == 1, 1, 0);
local ssabc13 = If (pabc13 == 1, 1, 0); local ssa13 = If (paa13 == 1, 1, 0); local ssab13 = If (pab13 == 1, 1, 0);
local ssabc14 = If (pabc14 == 1, 1, 0); local ssa14 = If (paa14 == 1, 1, 0); local ssab14 = If (pab14 == 1, 1, 0);
local ssabc15 = If (pabc15 == 1, 1, 0); local ssa15 = If (paa15 == 1, 1, 0); local ssab15 = If (pab15 == 1, 1, 0);
local ssabc16 = If (pabc16 == 1, 1, 0); local ssa16 = If (paa16 == 1, 1, 0); local ssab16 = If (pab16 == 1, 1, 0);
local ssabc17 = If (pabc17 == 1, 1, 0); local ssa17 = If (paa17 == 1, 1, 0); local ssab17 = If (pab17 == 1, 1, 0);
local ssabc18 = If (pabc18 == 1, 1, 0); local ssa18 = If (paa18 == 1, 1, 0); local ssab18 = If (pab18 == 1, 1, 0);
local ssabc19 = If (pabc19 == 1, 1, 0); local ssa19 = If (paa19 == 1, 1, 0); local ssab19 = If (pab19 == 1, 1, 0);
local ssabc20 = If (pabc20 == 1, 1, 0); local ssa20 = If (paa20 == 1, 1, 0); local ssab20 = If (pab20 == 1, 1, 0);
local ssabc21 = If (pabc21 == 1, 1, 0); local ssa21 = If (paa21 == 1, 1, 0); local ssab21 = If (pab21 == 1, 1, 0);
local ssabc22 = If (pabc22 == 1, 1, 0); local ssa22 = If (paa22 == 1, 1, 0); local ssab22 = If (pab22 == 1, 1, 0);
local ssabc23 = If (pabc23 == 1, 1, 0); local ssa23 = If (paa23 == 1, 1, 0); local ssab23 = If (pab23 == 1, 1, 0);
local ssabc24 = If (pabc24 == 1, 1, 0); local ssa24 = If (paa24 == 1, 1, 0); local ssab24 = If (pab24 == 1, 1, 0);

```

```

local ssabc25 = If (pabc25 == 1, 1, 0); local ssa25 = If (paa25 == 1, 1, 0); local ssab25 = If (pab25 == 1, 1, 0);
local ssabc26 = If (pabc26 == 1, 1, 0); local ssa26 = If (paa26 == 1, 1, 0); local ssab26 = If (pab26 == 1, 1, 0);
local ssabc27 = If (pabc27 == 1, 1, 0); local ssa27 = If (paa27 == 1, 1, 0); local ssab27 = If (pab27 == 1, 1, 0);
local ssabc28 = If (pabc28 == 1, 1, 0); local ssa28 = If (paa28 == 1, 1, 0); local ssab28 = If (pab28 == 1, 1, 0);
local ssabc29 = If (pabc29 == 1, 1, 0); local ssa29 = If (paa29 == 1, 1, 0); local ssab29 = If (pab29 == 1, 1, 0);
local ssabc30 = If (pabc30 == 1, 1, 0); local ssa30 = If (paa30 == 1, 1, 0); local ssab30 = If (pab30 == 1, 1, 0);
local ssabc31 = If (pabc31 == 1, 1, 0); local ssa31 = If (paa31 == 1, 1, 0); local ssab31 = If (pab31 == 1, 1, 0);
local ssabc32 = If (pabc32 == 1, 1, 0); local ssa32 = If (paa32 == 1, 1, 0); local ssab32 = If (pab32 == 1, 1, 0);
local ssabc33 = If (pabc33 == 1, 1, 0); local ssa33 = If (paa33 == 1, 1, 0); local ssab33 = If (pab33 == 1, 1, 0);
local ssabc34 = If (pabc34 == 1, 1, 0); local ssa34 = If (paa34 == 1, 1, 0); local ssab34 = If (pab34 == 1, 1, 0);
local ssabc35 = If (pabc35 == 1, 1, 0); local ssa35 = If (paa35 == 1, 1, 0); local ssab35 = If (pab35 == 1, 1, 0);
local ssabc36 = If (pabc36 == 1, 1, 0); local ssa36 = If (paa36 == 1, 1, 0); local ssab36 = If (pab36 == 1, 1, 0);
#####
#####
'SCALING FACTORS

```

'scaling of shifts to ABC stacking

```

local noraab2 = If (pabc2 == 1, 1, 3 * ssa2 + 0 * ssab2); local norbab2 = If (pabc2 == 1, 1, 1/(pabc2 + pab2
+ sabcab2)*ssa2 + (2*3)/(3-(3*paa2)+saa2)*ssab2);
local noraab3 = If (pabc3 == 1, 1, 3 * ssa3 + 0 * ssab3); local norbab3 = If (pabc3 == 1, 1, 1/(pabc3 + pab3
+ sabcab3)*ssa3 + (2*3)/(3-(3*paa3)+saa3)*ssab3);
local noraab4 = If (pabc4 == 1, 1, 3 * ssa4 + 0 * ssab4); local norbab4 = If (pabc4 == 1, 1, 1/(pabc4 + pab4
+ sabcab4)*ssa4 + (2*3)/(3-(3*paa4)+saa4)*ssab4);
local noraab5 = If (pabc5 == 1, 1, 3 * ssa5 + 0 * ssab5); local norbab5 = If (pabc5 == 1, 1, 1/(pabc5 + pab5
+ sabcab5)*ssa5 + (2*3)/(3-(3*paa5)+saa5)*ssab5);
local noraab6 = If (pabc6 == 1, 1, 3 * ssa6 + 0 * ssab6); local norbab6 = If (pabc6 == 1, 1, 1/(pabc6 + pab6
+ sabcab6)*ssa6 + (2*3)/(3-(3*paa6)+saa6)*ssab6);
local noraab7 = If (pabc7 == 1, 1, 3 * ssa7 + 0 * ssab7); local norbab7 = If (pabc7 == 1, 1, 1/(pabc7 + pab7
+ sabcab7)*ssa7 + (2*3)/(3-(3*paa7)+saa7)*ssab7);
local noraab8 = If (pabc8 == 1, 1, 3 * ssa8 + 0 * ssab8); local norbab8 = If (pabc8 == 1, 1, 1/(pabc8 + pab8
+ sabcab8)*ssa8 + (2*3)/(3-(3*paa8)+saa8)*ssab8);
local noraab9 = If (pabc9 == 1, 1, 3 * ssa9 + 0 * ssab9); local norbab9 = If (pabc9 == 1, 1, 1/(pabc9 + pab9
+ sabcab9)*ssa9 + (2*3)/(3-(3*paa9)+saa9)*ssab9);
local noraab10 = If (pabc10 == 1, 1, 3 * ssa10 + 0 * ssab10); local norbab10 = If (pabc10 == 1, 1,
1/(pabc10+pab10+sabcab10)*ssa10 + (2*3)/(3-(3*paa10)+saa10)*ssab10);
local noraab11 = If (pabc11 == 1, 1, 3 * ssa11 + 0 * ssab11); local norbab11 = If (pabc11 == 1, 1,
1/(pabc11+pab11+sabcab11)*ssa11 + (2*3)/(3-(3*paa11)+saa11)*ssab11);
local noraab12 = If (pabc12 == 1, 1, 3 * ssa12 + 0 * ssab12); local norbab12 = If (pabc12 == 1, 1,
1/(pabc12+pab12+sabcab12)*ssa12 + (2*3)/(3-(3*paa12)+saa12)*ssab12);
local noraab13 = If (pabc13 == 1, 1, 3 * ssa13 + 0 * ssab13); local norbab13 = If (pabc13 == 1, 1,
1/(pabc13+pab13+sabcab13)*ssa13 + (2*3)/(3-(3*paa13)+saa13)*ssab13);
local noraab14 = If (pabc14 == 1, 1, 3 * ssa14 + 0 * ssab14); local norbab14 = If (pabc14 == 1, 1,
1/(pabc14+pab14+sabcab14)*ssa14 + (2*3)/(3-(3*paa14)+saa14)*ssab14);
local noraab15 = If (pabc15 == 1, 1, 3 * ssa15 + 0 * ssab15); local norbab15 = If (pabc15 == 1, 1,
1/(pabc15+pab15+sabcab15)*ssa15 + (2*3)/(3-(3*paa15)+saa15)*ssab15);
local noraab16 = If (pabc16 == 1, 1, 3 * ssa16 + 0 * ssab16); local norbab16 = If (pabc16 == 1, 1,
1/(pabc16+pab16+sabcab16)*ssa16 + (2*3)/(3-(3*paa16)+saa16)*ssab16);
local noraab17 = If (pabc17 == 1, 1, 3 * ssa17 + 0 * ssab17); local norbab17 = If (pabc17 == 1, 1,
1/(pabc17+pab17+sabcab17)*ssa17 + (2*3)/(3-(3*paa17)+saa17)*ssab17);
local noraab18 = If (pabc18 == 1, 1, 3 * ssa18 + 0 * ssab18); local norbab18 = If (pabc18 == 1, 1,
1/(pabc18+pab18+sabcab18)*ssa18 + (2*3)/(3-(3*paa18)+saa18)*ssab18);
local noraab19 = If (pabc19 == 1, 1, 3 * ssa19 + 0 * ssab19); local norbab19 = If (pabc19 == 1, 1,
1/(pabc19+pab19+sabcab19)*ssa19 + (2*3)/(3-(3*paa19)+saa19)*ssab19);
local noraab20 = If (pabc20 == 1, 1, 3 * ssa20 + 0 * ssab20); local norbab20 = If (pabc20 == 1, 1,
1/(pabc20+pab20+sabcab20)*ssa20 + (2*3)/(3-(3*paa20)+saa20)*ssab20);
local noraab21 = If (pabc21 == 1, 1, 3 * ssa21 + 0 * ssab21); local norbab21 = If (pabc21 == 1, 1,
1/(pabc21+pab21+sabcab21)*ssa21 + (2*3)/(3-(3*paa21)+saa21)*ssab21);
local noraab22 = If (pabc22 == 1, 1, 3 * ssa22 + 0 * ssab22); local norbab22 = If (pabc22 == 1, 1,
1/(pabc22+pab22+sabcab22)*ssa22 + (2*3)/(3-(3*paa22)+saa22)*ssab22);
local noraab23 = If (pabc23 == 1, 1, 3 * ssa23 + 0 * ssab23); local norbab23 = If (pabc23 == 1, 1,
1/(pabc23+pab23+sabcab23)*ssa23 + (2*3)/(3-(3*paa23)+saa23)*ssab23);

```

```

local noraab24 = If (pabc24 == 1, 1, 3 * ssa24 + 0 * ssab24); local norbab24 = If (pabc24 == 1, 1,
1/(pabc24+pab24+sabcab24)*ssaa24 + (2*3)/(3-(3*paa24)+saa24)*ssab24);
local noraab25 = If (pabc25 == 1, 1, 3 * ssa25 + 0 * ssab25); local norbab25 = If (pabc25 == 1, 1,
1/(pabc25+pab25+sabcab25)*ssaa25 + (2*3)/(3-(3*paa25)+saa25)*ssab25);
local noraab26 = If (pabc26 == 1, 1, 3 * ssa26 + 0 * ssab26); local norbab26 = If (pabc26 == 1, 1,
1/(pabc26+pab26+sabcab26)*ssaa26 + (2*3)/(3-(3*paa26)+saa26)*ssab26);
local noraab27 = If (pabc27 == 1, 1, 3 * ssa27 + 0 * ssab27); local norbab27 = If (pabc27 == 1, 1,
1/(pabc27+pab27+sabcab27)*ssaa27 + (2*3)/(3-(3*paa27)+saa27)*ssab27);
local noraab28 = If (pabc28 == 1, 1, 3 * ssa28 + 0 * ssab28); local norbab28 = If (pabc28 == 1, 1,
1/(pabc28+pab28+sabcab28)*ssaa28 + (2*3)/(3-(3*paa28)+saa28)*ssab28);
local noraab29 = If (pabc29 == 1, 1, 3 * ssa29 + 0 * ssab29); local norbab29 = If (pabc29 == 1, 1,
1/(pabc29+pab29+sabcab29)*ssaa29 + (2*3)/(3-(3*paa29)+saa29)*ssab29);
local noraab30 = If (pabc30 == 1, 1, 3 * ssa30 + 0 * ssab30); local norbab30 = If (pabc30 == 1, 1,
1/(pabc30+pab30+sabcab30)*ssaa30 + (2*3)/(3-(3*paa30)+saa30)*ssab30);
local noraab31 = If (pabc31 == 1, 1, 3 * ssa31 + 0 * ssab31); local norbab31 = If (pabc31 == 1, 1,
1/(pabc31+pab31+sabcab31)*ssaa31 + (2*3)/(3-(3*paa31)+saa31)*ssab31);
local noraab32 = If (pabc32 == 1, 1, 3 * ssa32 + 0 * ssab32); local norbab32 = If (pabc32 == 1, 1,
1/(pabc32+pab32+sabcab32)*ssaa32 + (2*3)/(3-(3*paa32)+saa32)*ssab32);
local noraab33 = If (pabc33 == 1, 1, 3 * ssa33 + 0 * ssab33); local norbab33 = If (pabc33 == 1, 1,
1/(pabc33+pab33+sabcab33)*ssaa33 + (2*3)/(3-(3*paa33)+saa33)*ssab33);
local noraab34 = If (pabc34 == 1, 1, 3 * ssa34 + 0 * ssab34); local norbab34 = If (pabc34 == 1, 1,
1/(pabc34+pab34+sabcab34)*ssaa34 + (2*3)/(3-(3*paa34)+saa34)*ssab34);
local noraab35 = If (pabc35 == 1, 1, 3 * ssa35 + 0 * ssab35); local norbab35 = If (pabc35 == 1, 1,
1/(pabc35+pab35+sabcab35)*ssaa35 + (2*3)/(3-(3*paa35)+saa35)*ssab35);
local noraab36 = If (pabc36 == 1, 1, 3 * ssa36 + 0 * ssab36); local norbab36 = If (pabc36 == 1, 1,
1/(pabc36+pab36+sabcab36)*ssaa36 + (2*3)/(3-(3*paa36)+saa36)*ssab36);

```

```

-----
local norabc2 = If (pabc2 == 1, 1, 0 * ssa2 + 3 * ssab2 ); local norbbc2 = If (pabc2 == 1, 1, 2/(1-pab2 +sab2
)*ssaa2 - 1/(pabc2 + paa2 + sabcaa2 ) * ssab2 );
local norabc3 = If (pabc3 == 1, 1, 0 * ssa3 + 3 * ssab3 ); local norbbc3 = If (pabc3 == 1, 1, 2/(1-pab3 +sab3
)*ssaa3 - 1/(pabc3 + paa3 + sabcaa3 ) * ssab3 );
local norabc4 = If (pabc4 == 1, 1, 0 * ssa4 + 3 * ssab4 ); local norbbc4 = If (pabc4 == 1, 1, 2/(1-pab4 +sab4
)*ssaa4 - 1/(pabc4 + paa4 + sabcaa4 ) * ssab4 );
local norabc5 = If (pabc5 == 1, 1, 0 * ssa5 + 3 * ssab5 ); local norbbc5 = If (pabc5 == 1, 1, 2/(1-pab5 +sab5
)*ssaa5 - 1/(pabc5 + paa5 + sabcaa5 ) * ssab5 );
local norabc6 = If (pabc6 == 1, 1, 0 * ssa6 + 3 * ssab6 ); local norbbc6 = If (pabc6 == 1, 1, 2/(1-pab6 +sab6
)*ssaa6 - 1/(pabc6 + paa6 + sabcaa6 ) * ssab6 );
local norabc7 = If (pabc7 == 1, 1, 0 * ssa7 + 3 * ssab7 ); local norbbc7 = If (pabc7 == 1, 1, 2/(1-pab7 +sab7
)*ssaa7 - 1/(pabc7 + paa7 + sabcaa7 ) * ssab7 );
local norabc8 = If (pabc8 == 1, 1, 0 * ssa8 + 3 * ssab8 ); local norbbc8 = If (pabc8 == 1, 1, 2/(1-pab8 +sab8
)*ssaa8 - 1/(pabc8 + paa8 + sabcaa8 ) * ssab8 );
local norabc9 = If (pabc9 == 1, 1, 0 * ssa9 + 3 * ssab9 ); local norbbc9 = If (pabc9 == 1, 1, 2/(1-pab9 +sab9
)*ssaa9 - 1/(pabc9 + paa9 + sabcaa9 ) * ssab9 );
local norabc10 = If (pabc10 == 1, 1, 0 * ssa10 + 3 * ssab10); local norbbc10 = If (pabc10 == 1, 1, 2/(1-
pab10+sab10)*ssaa10 - 1/(pabc10 + paa10 + sabcaa10) * ssab10);
local norabc11 = If (pabc11 == 1, 1, 0 * ssa11 + 3 * ssab11); local norbbc11 = If (pabc11 == 1, 1, 2/(1-
pab11+sab11)*ssaa11 - 1/(pabc11 + paa11 + sabcaa11) * ssab11);
local norabc12 = If (pabc12 == 1, 1, 0 * ssa12 + 3 * ssab12); local norbbc12 = If (pabc12 == 1, 1, 2/(1-
pab12+sab12)*ssaa12 - 1/(pabc12 + paa12 + sabcaa12) * ssab12);
local norabc13 = If (pabc13 == 1, 1, 0 * ssa13 + 3 * ssab13); local norbbc13 = If (pabc13 == 1, 1, 2/(1-
pab13+sab13)*ssaa13 - 1/(pabc13 + paa13 + sabcaa13) * ssab13);
local norabc14 = If (pabc14 == 1, 1, 0 * ssa14 + 3 * ssab14); local norbbc14 = If (pabc14 == 1, 1, 2/(1-
pab14+sab14)*ssaa14 - 1/(pabc14 + paa14 + sabcaa14) * ssab14);
local norabc15 = If (pabc15 == 1, 1, 0 * ssa15 + 3 * ssab15); local norbbc15 = If (pabc15 == 1, 1, 2/(1-
pab15+sab15)*ssaa15 - 1/(pabc15 + paa15 + sabcaa15) * ssab15);
local norabc16 = If (pabc16 == 1, 1, 0 * ssa16 + 3 * ssab16); local norbbc16 = If (pabc16 == 1, 1, 2/(1-
pab16+sab16)*ssaa16 - 1/(pabc16 + paa16 + sabcaa16) * ssab16);
local norabc17 = If (pabc17 == 1, 1, 0 * ssa17 + 3 * ssab17); local norbbc17 = If (pabc17 == 1, 1, 2/(1-
pab17+sab17)*ssaa17 - 1/(pabc17 + paa17 + sabcaa17) * ssab17);

```

```

local norabc18 = If (pabc18 == 1, 1, 0 * ssaa18 + 3 * ssab18); local norbbc18 = If (pabc18 == 1, 1, 2/(1-
pab18+sab18)*ssaa18 - 1/(pabc18 + paa18 + sabcaa18) * ssab18);
local norabc19 = If (pabc19 == 1, 1, 0 * ssaa19 + 3 * ssab19); local norbbc19 = If (pabc19 == 1, 1, 2/(1-
pab19+sab19)*ssaa19 - 1/(pabc19 + paa19 + sabcaa19) * ssab19);
local norabc20 = If (pabc20 == 1, 1, 0 * ssaa20 + 3 * ssab20); local norbbc20 = If (pabc20 == 1, 1, 2/(1-
pab20+sab20)*ssaa20 - 1/(pabc20 + paa20 + sabcaa20) * ssab20);
local norabc21 = If (pabc21 == 1, 1, 0 * ssaa21 + 3 * ssab21); local norbbc21 = If (pabc21 == 1, 1, 2/(1-
pab21+sab21)*ssaa21 - 1/(pabc21 + paa21 + sabcaa21) * ssab21);
local norabc22 = If (pabc22 == 1, 1, 0 * ssaa22 + 3 * ssab22); local norbbc22 = If (pabc22 == 1, 1, 2/(1-
pab22+sab22)*ssaa22 - 1/(pabc22 + paa22 + sabcaa22) * ssab22);
local norabc23 = If (pabc23 == 1, 1, 0 * ssaa23 + 3 * ssab23); local norbbc23 = If (pabc23 == 1, 1, 2/(1-
pab23+sab23)*ssaa23 - 1/(pabc23 + paa23 + sabcaa23) * ssab23);
local norabc24 = If (pabc24 == 1, 1, 0 * ssaa24 + 3 * ssab24); local norbbc24 = If (pabc24 == 1, 1, 2/(1-
pab24+sab24)*ssaa24 - 1/(pabc24 + paa24 + sabcaa24) * ssab24);
local norabc25 = If (pabc25 == 1, 1, 0 * ssaa25 + 3 * ssab25); local norbbc25 = If (pabc25 == 1, 1, 2/(1-
pab25+sab25)*ssaa25 - 1/(pabc25 + paa25 + sabcaa25) * ssab25);
local norabc26 = If (pabc26 == 1, 1, 0 * ssaa26 + 3 * ssab26); local norbbc26 = If (pabc26 == 1, 1, 2/(1-
pab26+sab26)*ssaa26 - 1/(pabc26 + paa26 + sabcaa26) * ssab26);
local norabc27 = If (pabc27 == 1, 1, 0 * ssaa27 + 3 * ssab27); local norbbc27 = If (pabc27 == 1, 1, 2/(1-
pab27+sab27)*ssaa27 - 1/(pabc27 + paa27 + sabcaa27) * ssab27);
local norabc28 = If (pabc28 == 1, 1, 0 * ssaa28 + 3 * ssab28); local norbbc28 = If (pabc28 == 1, 1, 2/(1-
pab28+sab28)*ssaa28 - 1/(pabc28 + paa28 + sabcaa28) * ssab28);
local norabc29 = If (pabc29 == 1, 1, 0 * ssaa29 + 3 * ssab29); local norbbc29 = If (pabc29 == 1, 1, 2/(1-
pab29+sab29)*ssaa29 - 1/(pabc29 + paa29 + sabcaa29) * ssab29);
local norabc30 = If (pabc30 == 1, 1, 0 * ssaa30 + 3 * ssab30); local norbbc30 = If (pabc30 == 1, 1, 2/(1-
pab30+sab30)*ssaa30 - 1/(pabc30 + paa30 + sabcaa30) * ssab30);
local norabc31 = If (pabc31 == 1, 1, 0 * ssaa31 + 3 * ssab31); local norbbc31 = If (pabc31 == 1, 1, 2/(1-
pab31+sab31)*ssaa31 - 1/(pabc31 + paa31 + sabcaa31) * ssab31);
local norabc32 = If (pabc32 == 1, 1, 0 * ssaa32 + 3 * ssab32); local norbbc32 = If (pabc32 == 1, 1, 2/(1-
pab32+sab32)*ssaa32 - 1/(pabc32 + paa32 + sabcaa32) * ssab32);
local norabc33 = If (pabc33 == 1, 1, 0 * ssaa33 + 3 * ssab33); local norbbc33 = If (pabc33 == 1, 1, 2/(1-
pab33+sab33)*ssaa33 - 1/(pabc33 + paa33 + sabcaa33) * ssab33);
local norabc34 = If (pabc34 == 1, 1, 0 * ssaa34 + 3 * ssab34); local norbbc34 = If (pabc34 == 1, 1, 2/(1-
pab34+sab34)*ssaa34 - 1/(pabc34 + paa34 + sabcaa34) * ssab34);
local norabc35 = If (pabc35 == 1, 1, 0 * ssaa35 + 3 * ssab35); local norbbc35 = If (pabc35 == 1, 1, 2/(1-
pab35+sab35)*ssaa35 - 1/(pabc35 + paa35 + sabcaa35) * ssab35);
local norabc36 = If (pabc36 == 1, 1, 0 * ssaa36 + 3 * ssab36); local norbbc36 = If (pabc36 == 1, 1, 2/(1-
pab36+sab36)*ssaa36 - 1/(pabc36 + paa36 + sabcaa36) * ssab36);

```

```

-----
local noraca2 = If (pabc2 == 1, 1, 2* ssaa2 + 3 * ssab2 );
local noraca3 = If (pabc3 == 1, 1, 2* ssaa3 + 3 * ssab3 );
local noraca4 = If (pabc4 == 1, 1, 2* ssaa4 + 3 * ssab4 );
local noraca5 = If (pabc5 == 1, 1, 2* ssaa5 + 3 * ssab5 );
local noraca6 = If (pabc6 == 1, 1, 2* ssaa6 + 3 * ssab6 );
local noraca7 = If (pabc7 == 1, 1, 2* ssaa7 + 3 * ssab7 );
local noraca8 = If (pabc8 == 1, 1, 2* ssaa8 + 3 * ssab8 );
local noraca9 = If (pabc9 == 1, 1, 2* ssaa9 + 3 * ssab9 );
local noraca10 = If (pabc10 == 1, 1, 2* ssaa10 + 3 * ssab10);
local noraca11 = If (pabc11 == 1, 1, 2* ssaa11 + 3 * ssab11);
local noraca12 = If (pabc12 == 1, 1, 2* ssaa12 + 3 * ssab12);
local noraca13 = If (pabc13 == 1, 1, 2* ssaa13 + 3 * ssab13);
local noraca14 = If (pabc14 == 1, 1, 2* ssaa14 + 3 * ssab14);
local noraca15 = If (pabc15 == 1, 1, 2* ssaa15 + 3 * ssab15);
local noraca16 = If (pabc16 == 1, 1, 2* ssaa16 + 3 * ssab16);
local noraca17 = If (pabc17 == 1, 1, 2* ssaa17 + 3 * ssab17);
local noraca18 = If (pabc18 == 1, 1, 2* ssaa18 + 3 * ssab18);
local noraca19 = If (pabc19 == 1, 1, 2* ssaa19 + 3 * ssab19);
local noraca20 = If (pabc20 == 1, 1, 2* ssaa20 + 3 * ssab20);
local noraca21 = If (pabc21 == 1, 1, 2* ssaa21 + 3 * ssab21);
local noraca22 = If (pabc22 == 1, 1, 2* ssaa22 + 3 * ssab22);

```

```

local noraca23 = If (pabc23 == 1, 1, 2* ssaa23 + 3 * ssab23);
local noraca24 = If (pabc24 == 1, 1, 2* ssaa24 + 3 * ssab24);
local noraca25 = If (pabc25 == 1, 1, 2* ssaa25 + 3 * ssab25);
local noraca26 = If (pabc26 == 1, 1, 2* ssaa26 + 3 * ssab26);
local noraca27 = If (pabc27 == 1, 1, 2* ssaa27 + 3 * ssab27);
local noraca28 = If (pabc28 == 1, 1, 2* ssaa28 + 3 * ssab28);
local noraca29 = If (pabc29 == 1, 1, 2* ssaa29 + 3 * ssab29);
local noraca30 = If (pabc30 == 1, 1, 2* ssaa30 + 3 * ssab30);
local noraca31 = If (pabc31 == 1, 1, 2* ssaa31 + 3 * ssab31);
local noraca32 = If (pabc32 == 1, 1, 2* ssaa32 + 3 * ssab32);
local noraca33 = If (pabc33 == 1, 1, 2* ssaa33 + 3 * ssab33);
local noraca34 = If (pabc34 == 1, 1, 2* ssaa34 + 3 * ssab34);
local noraca35 = If (pabc35 == 1, 1, 2* ssaa35 + 3 * ssab35);
local noraca36 = If (pabc36 == 1, 1, 2* ssaa36 + 3 * ssab36);

```

```

-----
'-----
-----

```

'scaling of shifts to AA stacking

```

local noraaa2 = If (paa2 == 1, 1, 1.5);
local noraaa3 = If (paa3 == 1, 1, 1.5);
local noraaa4 = If (paa4 == 1, 1, 1.5);
local noraaa5 = If (paa5 == 1, 1, 1.5);
local noraaa6 = If (paa6 == 1, 1, 1.5);
local noraaa7 = If (paa7 == 1, 1, 1.5);
local noraaa8 = If (paa8 == 1, 1, 1.5);
local noraaa9 = If (paa9 == 1, 1, 1.5);
local noraaa10 = If (paa10 == 1, 1, 1.5);
local noraaa11 = If (paa11 == 1, 1, 1.5);
local noraaa12 = If (paa12 == 1, 1, 1.5);
local noraaa13 = If (paa13 == 1, 1, 1.5);
local noraaa14 = If (paa14 == 1, 1, 1.5);
local noraaa15 = If (paa15 == 1, 1, 1.5);
local noraaa16 = If (paa16 == 1, 1, 1.5);
local noraaa17 = If (paa17 == 1, 1, 1.5);
local noraaa18 = If (paa18 == 1, 1, 1.5);
local noraaa19 = If (paa19 == 1, 1, 1.5);
local noraaa20 = If (paa20 == 1, 1, 1.5);
local noraaa21 = If (paa21 == 1, 1, 1.5);
local noraaa22 = If (paa22 == 1, 1, 1.5);
local noraaa23 = If (paa23 == 1, 1, 1.5);
local noraaa24 = If (paa24 == 1, 1, 1.5);
local noraaa25 = If (paa25 == 1, 1, 1.5);
local noraaa26 = If (paa26 == 1, 1, 1.5);
local noraaa27 = If (paa27 == 1, 1, 1.5);
local noraaa28 = If (paa28 == 1, 1, 1.5);
local noraaa29 = If (paa29 == 1, 1, 1.5);
local noraaa30 = If (paa30 == 1, 1, 1.5);
local noraaa31 = If (paa31 == 1, 1, 1.5);
local noraaa32 = If (paa32 == 1, 1, 1.5);
local noraaa33 = If (paa33 == 1, 1, 1.5);
local noraaa34 = If (paa34 == 1, 1, 1.5);
local noraaa35 = If (paa35 == 1, 1, 1.5);
local noraaa36 = If (paa36 == 1, 1, 1.5);

```

```

-----
'-----
-----

```

```

local norabb2 = If (paa2 == 1, 1, 0 * ssabc2 + 3 * ssab2 ); local norbbb2 = If (paa2 == 1, 1, 2/(1-pab2 +sab2
)*ssabc2 + 1/(paa2 + pabc2 + sabcaa2 ) * ssab2 );
local norabb3 = If (paa3 == 1, 1, 0 * ssabc3 + 3 * ssab3 ); local norbbb3 = If (paa3 == 1, 1, 2/(1-pab3 +sab3
)*ssabc3 + 1/(paa3 + pabc3 + sabcaa3 ) * ssab3 );

```

local norabb4 = If (paa4 == 1, 1, 0 * ssabc4 + 3 * ssab4); local norbbb4 = If (paa4 == 1, 1, 2/(1-pab4 +sab4) *ssabc4 + 1/(paa4 + pabc4 + sabcaa4) * ssab4);
 local norabb5 = If (paa5 == 1, 1, 0 * ssabc5 + 3 * ssab5); local norbbb5 = If (paa5 == 1, 1, 2/(1-pab5 +sab5) *ssabc5 + 1/(paa5 + pabc5 + sabcaa5) * ssab5);
 local norabb6 = If (paa6 == 1, 1, 0 * ssabc6 + 3 * ssab6); local norbbb6 = If (paa6 == 1, 1, 2/(1-pab6 +sab6) *ssabc6 + 1/(paa6 + pabc6 + sabcaa6) * ssab6);
 local norabb7 = If (paa7 == 1, 1, 0 * ssabc7 + 3 * ssab7); local norbbb7 = If (paa7 == 1, 1, 2/(1-pab7 +sab7) *ssabc7 + 1/(paa7 + pabc7 + sabcaa7) * ssab7);
 local norabb8 = If (paa8 == 1, 1, 0 * ssabc8 + 3 * ssab8); local norbbb8 = If (paa8 == 1, 1, 2/(1-pab8 +sab8) *ssabc8 + 1/(paa8 + pabc8 + sabcaa8) * ssab8);
 local norabb9 = If (paa9 == 1, 1, 0 * ssabc9 + 3 * ssab9); local norbbb9 = If (paa9 == 1, 1, 2/(1-pab9 +sab9) *ssabc9 + 1/(paa9 + pabc9 + sabcaa9) * ssab9);
 local norabb10 = If (paa10 == 1, 1, 0 * ssabc10 + 3 * ssab10); local norbbb10 = If (paa10 == 1, 1, 2/(1-pab10+sab10)*ssabc10 + 1/(paa10 + pabc10 + sabcaa10) * ssab10);
 local norabb11 = If (paa11 == 1, 1, 0 * ssabc11 + 3 * ssab11); local norbbb11 = If (paa11 == 1, 1, 2/(1-pab11+sab11)*ssabc11 + 1/(paa11 + pabc11 + sabcaa11) * ssab11);
 local norabb12 = If (paa12 == 1, 1, 0 * ssabc12 + 3 * ssab12); local norbbb12 = If (paa12 == 1, 1, 2/(1-pab12+sab12)*ssabc12 + 1/(paa12 + pabc12 + sabcaa12) * ssab12);
 local norabb13 = If (paa13 == 1, 1, 0 * ssabc13 + 3 * ssab13); local norbbb13 = If (paa13 == 1, 1, 2/(1-pab13+sab13)*ssabc13 + 1/(paa13 + pabc13 + sabcaa13) * ssab13);
 local norabb14 = If (paa14 == 1, 1, 0 * ssabc14 + 3 * ssab14); local norbbb14 = If (paa14 == 1, 1, 2/(1-pab14+sab14)*ssabc14 + 1/(paa14 + pabc14 + sabcaa14) * ssab14);
 local norabb15 = If (paa15 == 1, 1, 0 * ssabc15 + 3 * ssab15); local norbbb15 = If (paa15 == 1, 1, 2/(1-pab15+sab15)*ssabc15 + 1/(paa15 + pabc15 + sabcaa15) * ssab15);
 local norabb16 = If (paa16 == 1, 1, 0 * ssabc16 + 3 * ssab16); local norbbb16 = If (paa16 == 1, 1, 2/(1-pab16+sab16)*ssabc16 + 1/(paa16 + pabc16 + sabcaa16) * ssab16);
 local norabb17 = If (paa17 == 1, 1, 0 * ssabc17 + 3 * ssab17); local norbbb17 = If (paa17 == 1, 1, 2/(1-pab17+sab17)*ssabc17 + 1/(paa17 + pabc17 + sabcaa17) * ssab17);
 local norabb18 = If (paa18 == 1, 1, 0 * ssabc18 + 3 * ssab18); local norbbb18 = If (paa18 == 1, 1, 2/(1-pab18+sab18)*ssabc18 + 1/(paa18 + pabc18 + sabcaa18) * ssab18);
 local norabb19 = If (paa19 == 1, 1, 0 * ssabc19 + 3 * ssab19); local norbbb19 = If (paa19 == 1, 1, 2/(1-pab19+sab19)*ssabc19 + 1/(paa19 + pabc19 + sabcaa19) * ssab19);
 local norabb20 = If (paa20 == 1, 1, 0 * ssabc20 + 3 * ssab20); local norbbb20 = If (paa20 == 1, 1, 2/(1-pab20+sab20)*ssabc20 + 1/(paa20 + pabc20 + sabcaa20) * ssab20);
 local norabb21 = If (paa21 == 1, 1, 0 * ssabc21 + 3 * ssab21); local norbbb21 = If (paa21 == 1, 1, 2/(1-pab21+sab21)*ssabc21 + 1/(paa21 + pabc21 + sabcaa21) * ssab21);
 local norabb22 = If (paa22 == 1, 1, 0 * ssabc22 + 3 * ssab22); local norbbb22 = If (paa22 == 1, 1, 2/(1-pab22+sab22)*ssabc22 + 1/(paa22 + pabc22 + sabcaa22) * ssab22);
 local norabb23 = If (paa23 == 1, 1, 0 * ssabc23 + 3 * ssab23); local norbbb23 = If (paa23 == 1, 1, 2/(1-pab23+sab23)*ssabc23 + 1/(paa23 + pabc23 + sabcaa23) * ssab23);
 local norabb24 = If (paa24 == 1, 1, 0 * ssabc24 + 3 * ssab24); local norbbb24 = If (paa24 == 1, 1, 2/(1-pab24+sab24)*ssabc24 + 1/(paa24 + pabc24 + sabcaa24) * ssab24);
 local norabb25 = If (paa25 == 1, 1, 0 * ssabc25 + 3 * ssab25); local norbbb25 = If (paa25 == 1, 1, 2/(1-pab25+sab25)*ssabc25 + 1/(paa25 + pabc25 + sabcaa25) * ssab25);
 local norabb26 = If (paa26 == 1, 1, 0 * ssabc26 + 3 * ssab26); local norbbb26 = If (paa26 == 1, 1, 2/(1-pab26+sab26)*ssabc26 + 1/(paa26 + pabc26 + sabcaa26) * ssab26);
 local norabb27 = If (paa27 == 1, 1, 0 * ssabc27 + 3 * ssab27); local norbbb27 = If (paa27 == 1, 1, 2/(1-pab27+sab27)*ssabc27 + 1/(paa27 + pabc27 + sabcaa27) * ssab27);
 local norabb28 = If (paa28 == 1, 1, 0 * ssabc28 + 3 * ssab28); local norbbb28 = If (paa28 == 1, 1, 2/(1-pab28+sab28)*ssabc28 + 1/(paa28 + pabc28 + sabcaa28) * ssab28);
 local norabb29 = If (paa29 == 1, 1, 0 * ssabc29 + 3 * ssab29); local norbbb29 = If (paa29 == 1, 1, 2/(1-pab29+sab29)*ssabc29 + 1/(paa29 + pabc29 + sabcaa29) * ssab29);
 local norabb30 = If (paa30 == 1, 1, 0 * ssabc30 + 3 * ssab30); local norbbb30 = If (paa30 == 1, 1, 2/(1-pab30+sab30)*ssabc30 + 1/(paa30 + pabc30 + sabcaa30) * ssab30);
 local norabb31 = If (paa31 == 1, 1, 0 * ssabc31 + 3 * ssab31); local norbbb31 = If (paa31 == 1, 1, 2/(1-pab31+sab31)*ssabc31 + 1/(paa31 + pabc31 + sabcaa31) * ssab31);
 local norabb32 = If (paa32 == 1, 1, 0 * ssabc32 + 3 * ssab32); local norbbb32 = If (paa32 == 1, 1, 2/(1-pab32+sab32)*ssabc32 + 1/(paa32 + pabc32 + sabcaa32) * ssab32);
 local norabb33 = If (paa33 == 1, 1, 0 * ssabc33 + 3 * ssab33); local norbbb33 = If (paa33 == 1, 1, 2/(1-pab33+sab33)*ssabc33 + 1/(paa33 + pabc33 + sabcaa33) * ssab33);

local norabb34 = If (paa34 == 1, 1, 0 * ssabc34 + 3 * ssab34); local norbbb34 = If (paa34 == 1, 1, 2/(1-pab34+sab34)*ssabc34 + 1/(paa34 + pabc34 + sabcaa34) * ssab34);
 local norabb35 = If (paa35 == 1, 1, 0 * ssabc35 + 3 * ssab35); local norbbb35 = If (paa35 == 1, 1, 2/(1-pab35+sab35)*ssabc35 + 1/(paa35 + pabc35 + sabcaa35) * ssab35);
 local norabb36 = If (paa36 == 1, 1, 0 * ssabc36 + 3 * ssab36); local norbbb36 = If (paa36 == 1, 1, 2/(1-pab36+sab36)*ssabc36 + 1/(paa36 + pabc36 + sabcaa36) * ssab36);

 local noracc2 = If (paa2 == 1, 1, 3 * ssabc2 + 0 * ssab2); local norbcc2 = If (paa2 == 1, 1, 1/(paa2 +pab2 +saaab2)*ssabc2 + 2/(1-pabc2 + sabc2) * ssab2);
 local noracc3 = If (paa3 == 1, 1, 3 * ssabc3 + 0 * ssab3); local norbcc3 = If (paa3 == 1, 1, 1/(paa3 +pab3 +saaab3)*ssabc3 + 2/(1-pabc3 + sabc3) * ssab3);
 local noracc4 = If (paa4 == 1, 1, 3 * ssabc4 + 0 * ssab4); local norbcc4 = If (paa4 == 1, 1, 1/(paa4 +pab4 +saaab4)*ssabc4 + 2/(1-pabc4 + sabc4) * ssab4);
 local noracc5 = If (paa5 == 1, 1, 3 * ssabc5 + 0 * ssab5); local norbcc5 = If (paa5 == 1, 1, 1/(paa5 +pab5 +saaab5)*ssabc5 + 2/(1-pabc5 + sabc5) * ssab5);
 local noracc6 = If (paa6 == 1, 1, 3 * ssabc6 + 0 * ssab6); local norbcc6 = If (paa6 == 1, 1, 1/(paa6 +pab6 +saaab6)*ssabc6 + 2/(1-pabc6 + sabc6) * ssab6);
 local noracc7 = If (paa7 == 1, 1, 3 * ssabc7 + 0 * ssab7); local norbcc7 = If (paa7 == 1, 1, 1/(paa7 +pab7 +saaab7)*ssabc7 + 2/(1-pabc7 + sabc7) * ssab7);
 local noracc8 = If (paa8 == 1, 1, 3 * ssabc8 + 0 * ssab8); local norbcc8 = If (paa8 == 1, 1, 1/(paa8 +pab8 +saaab8)*ssabc8 + 2/(1-pabc8 + sabc8) * ssab8);
 local noracc9 = If (paa9 == 1, 1, 3 * ssabc9 + 0 * ssab9); local norbcc9 = If (paa9 == 1, 1, 1/(paa9 +pab9 +saaab9)*ssabc9 + 2/(1-pabc9 + sabc9) * ssab9);
 local noracc10 = If (paa10 == 1, 1, 3 * ssabc10 + 0 * ssab10); local norbcc10 = If (paa10 == 1, 1, 1/(paa10+pab10+saaab10)*ssabc10 + 2/(1-pabc10 + sabc10) * ssab10);
 local noracc11 = If (paa11 == 1, 1, 3 * ssabc11 + 0 * ssab11); local norbcc11 = If (paa11 == 1, 1, 1/(paa11+pab11+saaab11)*ssabc11 + 2/(1-pabc11 + sabc11) * ssab11);
 local noracc12 = If (paa12 == 1, 1, 3 * ssabc12 + 0 * ssab12); local norbcc12 = If (paa12 == 1, 1, 1/(paa12+pab12+saaab12)*ssabc12 + 2/(1-pabc12 + sabc12) * ssab12);
 local noracc13 = If (paa13 == 1, 1, 3 * ssabc13 + 0 * ssab13); local norbcc13 = If (paa13 == 1, 1, 1/(paa13+pab13+saaab13)*ssabc13 + 2/(1-pabc13 + sabc13) * ssab13);
 local noracc14 = If (paa14 == 1, 1, 3 * ssabc14 + 0 * ssab14); local norbcc14 = If (paa14 == 1, 1, 1/(paa14+pab14+saaab14)*ssabc14 + 2/(1-pabc14 + sabc14) * ssab14);
 local noracc15 = If (paa15 == 1, 1, 3 * ssabc15 + 0 * ssab15); local norbcc15 = If (paa15 == 1, 1, 1/(paa15+pab15+saaab15)*ssabc15 + 2/(1-pabc15 + sabc15) * ssab15);
 local noracc16 = If (paa16 == 1, 1, 3 * ssabc16 + 0 * ssab16); local norbcc16 = If (paa16 == 1, 1, 1/(paa16+pab16+saaab16)*ssabc16 + 2/(1-pabc16 + sabc16) * ssab16);
 local noracc17 = If (paa17 == 1, 1, 3 * ssabc17 + 0 * ssab17); local norbcc17 = If (paa17 == 1, 1, 1/(paa17+pab17+saaab17)*ssabc17 + 2/(1-pabc17 + sabc17) * ssab17);
 local noracc18 = If (paa18 == 1, 1, 3 * ssabc18 + 0 * ssab18); local norbcc18 = If (paa18 == 1, 1, 1/(paa18+pab18+saaab18)*ssabc18 + 2/(1-pabc18 + sabc18) * ssab18);
 local noracc19 = If (paa19 == 1, 1, 3 * ssabc19 + 0 * ssab19); local norbcc19 = If (paa19 == 1, 1, 1/(paa19+pab19+saaab19)*ssabc19 + 2/(1-pabc19 + sabc19) * ssab19);
 local noracc20 = If (paa20 == 1, 1, 3 * ssabc20 + 0 * ssab20); local norbcc20 = If (paa20 == 1, 1, 1/(paa20+pab20+saaab20)*ssabc20 + 2/(1-pabc20 + sabc20) * ssab20);
 local noracc21 = If (paa21 == 1, 1, 3 * ssabc21 + 0 * ssab21); local norbcc21 = If (paa21 == 1, 1, 1/(paa21+pab21+saaab21)*ssabc21 + 2/(1-pabc21 + sabc21) * ssab21);
 local noracc22 = If (paa22 == 1, 1, 3 * ssabc22 + 0 * ssab22); local norbcc22 = If (paa22 == 1, 1, 1/(paa22+pab22+saaab22)*ssabc22 + 2/(1-pabc22 + sabc22) * ssab22);
 local noracc23 = If (paa23 == 1, 1, 3 * ssabc23 + 0 * ssab23); local norbcc23 = If (paa23 == 1, 1, 1/(paa23+pab23+saaab23)*ssabc23 + 2/(1-pabc23 + sabc23) * ssab23);
 local noracc24 = If (paa24 == 1, 1, 3 * ssabc24 + 0 * ssab24); local norbcc24 = If (paa24 == 1, 1, 1/(paa24+pab24+saaab24)*ssabc24 + 2/(1-pabc24 + sabc24) * ssab24);
 local noracc25 = If (paa25 == 1, 1, 3 * ssabc25 + 0 * ssab25); local norbcc25 = If (paa25 == 1, 1, 1/(paa25+pab25+saaab25)*ssabc25 + 2/(1-pabc25 + sabc25) * ssab25);
 local noracc26 = If (paa26 == 1, 1, 3 * ssabc26 + 0 * ssab26); local norbcc26 = If (paa26 == 1, 1, 1/(paa26+pab26+saaab26)*ssabc26 + 2/(1-pabc26 + sabc26) * ssab26);
 local noracc27 = If (paa27 == 1, 1, 3 * ssabc27 + 0 * ssab27); local norbcc27 = If (paa27 == 1, 1, 1/(paa27+pab27+saaab27)*ssabc27 + 2/(1-pabc27 + sabc27) * ssab27);

```

local noracc28 = If (paa28 == 1, 1, 3 * ssabc28 + 0 * ssab28); local norbcc28 = If (paa28 == 1, 1,
1/(paa28+pab28+saaab28)*ssabc28 + 2/(1-pabc28 + sabc28) * ssab28);
local noracc29 = If (paa29 == 1, 1, 3 * ssabc29 + 0 * ssab29); local norbcc29 = If (paa29 == 1, 1,
1/(paa29+pab29+saaab29)*ssabc29 + 2/(1-pabc29 + sabc29) * ssab29);
local noracc30 = If (paa30 == 1, 1, 3 * ssabc30 + 0 * ssab30); local norbcc30 = If (paa30 == 1, 1,
1/(paa30+pab30+saaab30)*ssabc30 + 2/(1-pabc30 + sabc30) * ssab30);
local noracc31 = If (paa31 == 1, 1, 3 * ssabc31 + 0 * ssab31); local norbcc31 = If (paa31 == 1, 1,
1/(paa31+pab31+saaab31)*ssabc31 + 2/(1-pabc31 + sabc31) * ssab31);
local noracc32 = If (paa32 == 1, 1, 3 * ssabc32 + 0 * ssab32); local norbcc32 = If (paa32 == 1, 1,
1/(paa32+pab32+saaab32)*ssabc32 + 2/(1-pabc32 + sabc32) * ssab32);
local noracc33 = If (paa33 == 1, 1, 3 * ssabc33 + 0 * ssab33); local norbcc33 = If (paa33 == 1, 1,
1/(paa33+pab33+saaab33)*ssabc33 + 2/(1-pabc33 + sabc33) * ssab33);
local noracc34 = If (paa34 == 1, 1, 3 * ssabc34 + 0 * ssab34); local norbcc34 = If (paa34 == 1, 1,
1/(paa34+pab34+saaab34)*ssabc34 + 2/(1-pabc34 + sabc34) * ssab34);
local noracc35 = If (paa35 == 1, 1, 3 * ssabc35 + 0 * ssab35); local norbcc35 = If (paa35 == 1, 1,
1/(paa35+pab35+saaab35)*ssabc35 + 2/(1-pabc35 + sabc35) * ssab35);
local noracc36 = If (paa36 == 1, 1, 3 * ssabc36 + 0 * ssab36); local norbcc36 = If (paa36 == 1, 1,
1/(paa36+pab36+saaab36)*ssabc36 + 2/(1-pabc36 + sabc36) * ssab36);

```

'scaling of shifts to CBA stacking

```

local noraac2 = If (pab2 == 1, 1, 0); local norbac2 = If (pab2 == 1, 1, (2*1)/(1-paa2 +saa2)*ssabc2 +1/(pabc2
+pab2 +sabcab2)*ssaa2);
local noraac3 = If (pab3 == 1, 1, 0); local norbac3 = If (pab3 == 1, 1, (2*1)/(1-paa3 +saa3)*ssabc3 +1/(pabc3
+pab3 +sabcab3)*ssaa3);
local noraac4 = If (pab4 == 1, 1, 0); local norbac4 = If (pab4 == 1, 1, (2*1)/(1-paa4 +saa4)*ssabc4 +1/(pabc4
+pab4 +sabcab4)*ssaa4);
local noraac5 = If (pab5 == 1, 1, 0); local norbac5 = If (pab5 == 1, 1, (2*1)/(1-paa5 +saa5)*ssabc5 +1/(pabc5
+pab5 +sabcab5)*ssaa5);
local noraac6 = If (pab6 == 1, 1, 0); local norbac6 = If (pab6 == 1, 1, (2*1)/(1-paa6 +saa6)*ssabc6 +1/(pabc6
+pab6 +sabcab6)*ssaa6);
local noraac7 = If (pab7 == 1, 1, 0); local norbac7 = If (pab7 == 1, 1, (2*1)/(1-paa7 +saa7)*ssabc7 +1/(pabc7
+pab7 +sabcab7)*ssaa7);
local noraac8 = If (pab8 == 1, 1, 0); local norbac8 = If (pab8 == 1, 1, (2*1)/(1-paa8 +saa8)*ssabc8 +1/(pabc8
+pab8 +sabcab8)*ssaa8);
local noraac9 = If (pab9 == 1, 1, 0); local norbac9 = If (pab9 == 1, 1, (2*1)/(1-paa9 +saa9)*ssabc9 +1/(pabc9
+pab9 +sabcab9)*ssaa9);
local noraac10 = If (pab10 == 1, 1, 0); local norbac10 = If (pab10 == 1, 1, (2*1)/(1-
paa10+saa10)*ssabc10+1/(pabc10+pab10+sabcab10)*ssaa10);
local noraac11 = If (pab11 == 1, 1, 0); local norbac11 = If (pab11 == 1, 1, (2*1)/(1-
paa11+saa11)*ssabc11+1/(pabc11+pab11+sabcab11)*ssaa11);
local noraac12 = If (pab12 == 1, 1, 0); local norbac12 = If (pab12 == 1, 1, (2*1)/(1-
paa12+saa12)*ssabc12+1/(pabc12+pab12+sabcab12)*ssaa12);
local noraac13 = If (pab13 == 1, 1, 0); local norbac13 = If (pab13 == 1, 1, (2*1)/(1-
paa13+saa13)*ssabc13+1/(pabc13+pab13+sabcab13)*ssaa13);
local noraac14 = If (pab14 == 1, 1, 0); local norbac14 = If (pab14 == 1, 1, (2*1)/(1-
paa14+saa14)*ssabc14+1/(pabc14+pab14+sabcab14)*ssaa14);
local noraac15 = If (pab15 == 1, 1, 0); local norbac15 = If (pab15 == 1, 1, (2*1)/(1-
paa15+saa15)*ssabc15+1/(pabc15+pab15+sabcab15)*ssaa15);
local noraac16 = If (pab16 == 1, 1, 0); local norbac16 = If (pab16 == 1, 1, (2*1)/(1-
paa16+saa16)*ssabc16+1/(pabc16+pab16+sabcab16)*ssaa16);
local noraac17 = If (pab17 == 1, 1, 0); local norbac17 = If (pab17 == 1, 1, (2*1)/(1-
paa17+saa17)*ssabc17+1/(pabc17+pab17+sabcab17)*ssaa17);
local noraac18 = If (pab18 == 1, 1, 0); local norbac18 = If (pab18 == 1, 1, (2*1)/(1-
paa18+saa18)*ssabc18+1/(pabc18+pab18+sabcab18)*ssaa18);
local noraac19 = If (pab19 == 1, 1, 0); local norbac19 = If (pab19 == 1, 1, (2*1)/(1-
paa19+saa19)*ssabc19+1/(pabc19+pab19+sabcab19)*ssaa19);
local noraac20 = If (pab20 == 1, 1, 0); local norbac20 = If (pab20 == 1, 1, (2*1)/(1-
paa20+saa20)*ssabc20+1/(pabc20+pab20+sabcab20)*ssaa20);

```



```

local noraac21 = If (pab21 == 1, 1, 0); local norbac21 = If (pab21 == 1, 1, (2*1)/(1-
paa21+saa21)*ssabc21+1/(pabc21+pab21+sabcab21)*ssaa21);
local noraac22 = If (pab22 == 1, 1, 0); local norbac22 = If (pab22 == 1, 1, (2*1)/(1-
paa22+saa22)*ssabc22+1/(pabc22+pab22+sabcab22)*ssaa22);
local noraac23 = If (pab23 == 1, 1, 0); local norbac23 = If (pab23 == 1, 1, (2*1)/(1-
paa23+saa23)*ssabc23+1/(pabc23+pab23+sabcab23)*ssaa23);
local noraac24 = If (pab24 == 1, 1, 0); local norbac24 = If (pab24 == 1, 1, (2*1)/(1-
paa24+saa24)*ssabc24+1/(pabc24+pab24+sabcab24)*ssaa24);
local noraac25 = If (pab25 == 1, 1, 0); local norbac25 = If (pab25 == 1, 1, (2*1)/(1-
paa25+saa25)*ssabc25+1/(pabc25+pab25+sabcab25)*ssaa25);
local noraac26 = If (pab26 == 1, 1, 0); local norbac26 = If (pab26 == 1, 1, (2*1)/(1-
paa26+saa26)*ssabc26+1/(pabc26+pab26+sabcab26)*ssaa26);
local noraac27 = If (pab27 == 1, 1, 0); local norbac27 = If (pab27 == 1, 1, (2*1)/(1-
paa27+saa27)*ssabc27+1/(pabc27+pab27+sabcab27)*ssaa27);
local noraac28 = If (pab28 == 1, 1, 0); local norbac28 = If (pab28 == 1, 1, (2*1)/(1-
paa28+saa28)*ssabc28+1/(pabc28+pab28+sabcab28)*ssaa28);
local noraac29 = If (pab29 == 1, 1, 0); local norbac29 = If (pab29 == 1, 1, (2*1)/(1-
paa29+saa29)*ssabc29+1/(pabc29+pab29+sabcab29)*ssaa29);
local noraac30 = If (pab30 == 1, 1, 0); local norbac30 = If (pab30 == 1, 1, (2*1)/(1-
paa30+saa30)*ssabc30+1/(pabc30+pab30+sabcab30)*ssaa30);
local noraac31 = If (pab31 == 1, 1, 0); local norbac31 = If (pab31 == 1, 1, (2*1)/(1-
paa31+saa31)*ssabc31+1/(pabc31+pab31+sabcab31)*ssaa31);
local noraac32 = If (pab32 == 1, 1, 0); local norbac32 = If (pab32 == 1, 1, (2*1)/(1-
paa32+saa32)*ssabc32+1/(pabc32+pab32+sabcab32)*ssaa32);
local noraac33 = If (pab33 == 1, 1, 0); local norbac33 = If (pab33 == 1, 1, (2*1)/(1-
paa33+saa33)*ssabc33+1/(pabc33+pab33+sabcab33)*ssaa33);
local noraac34 = If (pab34 == 1, 1, 0); local norbac34 = If (pab34 == 1, 1, (2*1)/(1-
paa34+saa34)*ssabc34+1/(pabc34+pab34+sabcab34)*ssaa34);
local noraac35 = If (pab35 == 1, 1, 0); local norbac35 = If (pab35 == 1, 1, (2*1)/(1-
paa35+saa35)*ssabc35+1/(pabc35+pab35+sabcab35)*ssaa35);
local noraac36 = If (pab36 == 1, 1, 0); local norbac36 = If (pab35 == 1, 1, (2*1)/(1-
paa36+saa36)*ssabc36+1/(pabc36+pab36+sabcab36)*ssaa36);
'-----

```

```

local noraba2 = If (pab2 == 1, 1, 1.5);
local noraba3 = If (pab3 == 1, 1, 1.5);
local noraba4 = If (pab4 == 1, 1, 1.5);
local noraba5 = If (pab5 == 1, 1, 1.5);
local noraba6 = If (pab6 == 1, 1, 1.5);
local noraba7 = If (pab7 == 1, 1, 1.5);
local noraba8 = If (pab8 == 1, 1, 1.5);
local noraba9 = If (pab9 == 1, 1, 1.5);
local noraba10 = If (pab10 == 1, 1, 1.5);
local noraba11 = If (pab11 == 1, 1, 1.5);
local noraba12 = If (pab12 == 1, 1, 1.5);
local noraba13 = If (pab13 == 1, 1, 1.5);
local noraba14 = If (pab14 == 1, 1, 1.5);
local noraba15 = If (pab15 == 1, 1, 1.5);
local noraba16 = If (pab16 == 1, 1, 1.5);
local noraba17 = If (pab17 == 1, 1, 1.5);
local noraba18 = If (pab18 == 1, 1, 1.5);
local noraba19 = If (pab19 == 1, 1, 1.5);
local noraba20 = If (pab20 == 1, 1, 1.5);
local noraba21 = If (pab21 == 1, 1, 1.5);
local noraba22 = If (pab22 == 1, 1, 1.5);
local noraba23 = If (pab23 == 1, 1, 1.5);
local noraba24 = If (pab24 == 1, 1, 1.5);
local noraba25 = If (pab25 == 1, 1, 1.5);
local noraba26 = If (pab26 == 1, 1, 1.5);
local noraba27 = If (pab27 == 1, 1, 1.5);
local noraba28 = If (pab28 == 1, 1, 1.5);

```

local noraba29 = If (pab29 == 1, 1, 1.5);
 local noraba30 = If (pab30 == 1, 1, 1.5);
 local noraba31 = If (pab31 == 1, 1, 1.5);
 local noraba32 = If (pab32 == 1, 1, 1.5);
 local noraba33 = If (pab33 == 1, 1, 1.5);
 local noraba34 = If (pab34 == 1, 1, 1.5);
 local noraba35 = If (pab35 == 1, 1, 1.5);
 local noraba36 = If (pab36 == 1, 1, 1.5);

local noracb2 = If (pab2 == 1, 1, 3); local norbcb2 = If (pab2 == 1, 1, 1/(pab2 + paa2 + saaab2)*ssabc2
 +(2*3)/(3-(3*pabc2) + sabc2)*ssaa2);
 local noracb3 = If (pab3 == 1, 1, 3); local norbcb3 = If (pab3 == 1, 1, 1/(pab3 + paa3 + saaab3)*ssabc3
 +(2*3)/(3-(3*pabc3) + sabc3)*ssaa3);
 local noracb4 = If (pab4 == 1, 1, 3); local norbcb4 = If (pab4 == 1, 1, 1/(pab4 + paa4 + saaab4)*ssabc4
 +(2*3)/(3-(3*pabc4) + sabc4)*ssaa4);
 local noracb5 = If (pab5 == 1, 1, 3); local norbcb5 = If (pab5 == 1, 1, 1/(pab5 + paa5 + saaab5)*ssabc5
 +(2*3)/(3-(3*pabc5) + sabc5)*ssaa5);
 local noracb6 = If (pab6 == 1, 1, 3); local norbcb6 = If (pab6 == 1, 1, 1/(pab6 + paa6 + saaab6)*ssabc6
 +(2*3)/(3-(3*pabc6) + sabc6)*ssaa6);
 local noracb7 = If (pab7 == 1, 1, 3); local norbcb7 = If (pab7 == 1, 1, 1/(pab7 + paa7 + saaab7)*ssabc7
 +(2*3)/(3-(3*pabc7) + sabc7)*ssaa7);
 local noracb8 = If (pab8 == 1, 1, 3); local norbcb8 = If (pab8 == 1, 1, 1/(pab8 + paa8 + saaab8)*ssabc8
 +(2*3)/(3-(3*pabc8) + sabc8)*ssaa8);
 local noracb9 = If (pab9 == 1, 1, 3); local norbcb9 = If (pab9 == 1, 1, 1/(pab9 + paa9 + saaab9)*ssabc9
 +(2*3)/(3-(3*pabc9) + sabc9)*ssaa9);
 local noracb10 = If (pab10 == 1, 1, 3); local norbcb10 = If (pab10 == 1, 1, 1/(pab10 + paa10 +
 saaab10)*ssabc10+(2*3)/(3-(3*pabc10) + sabc10)*ssaa10);
 local noracb11 = If (pab11 == 1, 1, 3); local norbcb11 = If (pab11 == 1, 1, 1/(pab11 + paa11 +
 saaab11)*ssabc11+(2*3)/(3-(3*pabc11) + sabc11)*ssaa11);
 local noracb12 = If (pab12 == 1, 1, 3); local norbcb12 = If (pab12 == 1, 1, 1/(pab12 + paa12 +
 saaab12)*ssabc12+(2*3)/(3-(3*pabc12) + sabc12)*ssaa12);
 local noracb13 = If (pab13 == 1, 1, 3); local norbcb13 = If (pab13 == 1, 1, 1/(pab13 + paa13 +
 saaab13)*ssabc13+(2*3)/(3-(3*pabc13) + sabc13)*ssaa13);
 local noracb14 = If (pab14 == 1, 1, 3); local norbcb14 = If (pab14 == 1, 1, 1/(pab14 + paa14 +
 saaab14)*ssabc14+(2*3)/(3-(3*pabc14) + sabc14)*ssaa14);
 local noracb15 = If (pab15 == 1, 1, 3); local norbcb15 = If (pab15 == 1, 1, 1/(pab15 + paa15 +
 saaab15)*ssabc15+(2*3)/(3-(3*pabc15) + sabc15)*ssaa15);
 local noracb16 = If (pab16 == 1, 1, 3); local norbcb16 = If (pab16 == 1, 1, 1/(pab16 + paa16 +
 saaab16)*ssabc16+(2*3)/(3-(3*pabc16) + sabc16)*ssaa16);
 local noracb17 = If (pab17 == 1, 1, 3); local norbcb17 = If (pab17 == 1, 1, 1/(pab17 + paa17 +
 saaab17)*ssabc17+(2*3)/(3-(3*pabc17) + sabc17)*ssaa17);
 local noracb18 = If (pab18 == 1, 1, 3); local norbcb18 = If (pab18 == 1, 1, 1/(pab18 + paa18 +
 saaab18)*ssabc18+(2*3)/(3-(3*pabc18) + sabc18)*ssaa18);
 local noracb19 = If (pab19 == 1, 1, 3); local norbcb19 = If (pab19 == 1, 1, 1/(pab19 + paa19 +
 saaab19)*ssabc19+(2*3)/(3-(3*pabc19) + sabc19)*ssaa19);
 local noracb20 = If (pab20 == 1, 1, 3); local norbcb20 = If (pab20 == 1, 1, 1/(pab20 + paa20 +
 saaab20)*ssabc20+(2*3)/(3-(3*pabc20) + sabc20)*ssaa20);
 local noracb21 = If (pab21 == 1, 1, 3); local norbcb21 = If (pab21 == 1, 1, 1/(pab21 + paa21 +
 saaab21)*ssabc21+(2*3)/(3-(3*pabc21) + sabc21)*ssaa21);
 local noracb22 = If (pab22 == 1, 1, 3); local norbcb22 = If (pab22 == 1, 1, 1/(pab22 + paa22 +
 saaab22)*ssabc22+(2*3)/(3-(3*pabc22) + sabc22)*ssaa22);
 local noracb23 = If (pab23 == 1, 1, 3); local norbcb23 = If (pab23 == 1, 1, 1/(pab23 + paa23 +
 saaab23)*ssabc23+(2*3)/(3-(3*pabc23) + sabc23)*ssaa23);
 local noracb24 = If (pab24 == 1, 1, 3); local norbcb24 = If (pab24 == 1, 1, 1/(pab24 + paa24 +
 saaab24)*ssabc24+(2*3)/(3-(3*pabc24) + sabc24)*ssaa24);
 local noracb25 = If (pab25 == 1, 1, 3); local norbcb25 = If (pab25 == 1, 1, 1/(pab25 + paa25 +
 saaab25)*ssabc25+(2*3)/(3-(3*pabc25) + sabc25)*ssaa25);
 local noracb26 = If (pab26 == 1, 1, 3); local norbcb26 = If (pab26 == 1, 1, 1/(pab26 + paa26 +
 saaab26)*ssabc26+(2*3)/(3-(3*pabc26) + sabc26)*ssaa26);

```

local noracb27 = If (pab27 == 1, 1, 3); local norbcb27 = If (pab27 == 1, 1, 1/(pab27 + paa27 +
saaab27)*ssabc27+(2*3)/(3-(3*pabc27) + sabc27)*ssaa27);
local noracb28 = If (pab28 == 1, 1, 3); local norbcb28 = If (pab28 == 1, 1, 1/(pab28 + paa28 +
saaab28)*ssabc28+(2*3)/(3-(3*pabc28) + sabc28)*ssaa28);
local noracb29 = If (pab29 == 1, 1, 3); local norbcb29 = If (pab29 == 1, 1, 1/(pab29 + paa29 +
saaab29)*ssabc29+(2*3)/(3-(3*pabc29) + sabc29)*ssaa29);
local noracb30 = If (pab30 == 1, 1, 3); local norbcb30 = If (pab30 == 1, 1, 1/(pab30 + paa30 +
saaab30)*ssabc30+(2*3)/(3-(3*pabc30) + sabc30)*ssaa30);
local noracb31 = If (pab31 == 1, 1, 3); local norbcb31 = If (pab31 == 1, 1, 1/(pab31 + paa31 +
saaab31)*ssabc31+(2*3)/(3-(3*pabc31) + sabc31)*ssaa31);
local noracb32 = If (pab32 == 1, 1, 3); local norbcb32 = If (pab32 == 1, 1, 1/(pab32 + paa32 +
saaab32)*ssabc32+(2*3)/(3-(3*pabc32) + sabc32)*ssaa32);
local noracb33 = If (pab33 == 1, 1, 3); local norbcb33 = If (pab33 == 1, 1, 1/(pab33 + paa33 +
saaab33)*ssabc33+(2*3)/(3-(3*pabc33) + sabc33)*ssaa33);
local noracb34 = If (pab34 == 1, 1, 3); local norbcb34 = If (pab34 == 1, 1, 1/(pab34 + paa34 +
saaab34)*ssabc34+(2*3)/(3-(3*pabc34) + sabc34)*ssaa34);
local noracb35 = If (pab35 == 1, 1, 3); local norbcb35 = If (pab35 == 1, 1, 1/(pab35 + paa35 +
saaab35)*ssabc35+(2*3)/(3-(3*pabc35) + sabc35)*ssaa35);
local noracb36 = If (pab36 == 1, 1, 3); local norbcb36 = If (pab36 == 1, 1, 1/(pab36 + paa36 +
saaab36)*ssabc36+(2*3)/(3-(3*pabc36) + sabc36)*ssaa36);

```

```

#####
#####

```

'SELECTION OF THE DECISION PARAMETERS

'decision parameter deca

```

prm deca2 = 1 +pabc2 * ((noraab2 * ab2) - (norabc2 * bc2) + 2*(noraca2 * ca2)) +paa2 * ((2* noraaa2 *
aa2) + (norabb2 * bb2) - (noracc2 * cc2)) +pab2 * ((-1* noraac2 * ac2) + (2* noraba2 * ba2) + (noracb2 *
cb2));
prm deca3 = 1 +pabc3 * ((noraab3 * ab3) - (norabc3 * bc3) + 2*(noraca3 * ca3)) +paa3 * ((2* noraaa3 *
aa3) + (norabb3 * bb3) - (noracc3 * cc3)) +pab3 * ((-1* noraac3 * ac3) + (2* noraba3 * ba3) + (noracb3 *
cb3));
prm deca4 = 1 +pabc4 * ((noraab4 * ab4) - (norabc4 * bc4) + 2*(noraca4 * ca4)) +paa4 * ((2* noraaa4 *
aa4) + (norabb4 * bb4) - (noracc4 * cc4)) +pab4 * ((-1* noraac4 * ac4) + (2* noraba4 * ba4) + (noracb4 *
cb4));
prm deca5 = 1 +pabc5 * ((noraab5 * ab5) - (norabc5 * bc5) + 2*(noraca5 * ca5)) +paa5 * ((2* noraaa5 *
aa5) + (norabb5 * bb5) - (noracc5 * cc5)) +pab5 * ((-1* noraac5 * ac5) + (2* noraba5 * ba5) + (noracb5 *
cb5));
prm deca6 = 1 +pabc6 * ((noraab6 * ab6) - (norabc6 * bc6) + 2*(noraca6 * ca6)) +paa6 * ((2* noraaa6 *
aa6) + (norabb6 * bb6) - (noracc6 * cc6)) +pab6 * ((-1* noraac6 * ac6) + (2* noraba6 * ba6) + (noracb6 *
cb6));
prm deca7 = 1 +pabc7 * ((noraab7 * ab7) - (norabc7 * bc7) + 2*(noraca7 * ca7)) +paa7 * ((2* noraaa7 *
aa7) + (norabb7 * bb7) - (noracc7 * cc7)) +pab7 * ((-1* noraac7 * ac7) + (2* noraba7 * ba7) + (noracb7 *
cb7));
prm deca8 = 1 +pabc8 * ((noraab8 * ab8) - (norabc8 * bc8) + 2*(noraca8 * ca8)) +paa8 * ((2* noraaa8 *
aa8) + (norabb8 * bb8) - (noracc8 * cc8)) +pab8 * ((-1* noraac8 * ac8) + (2* noraba8 * ba8) + (noracb8 *
cb8));
prm deca9 = 1 +pabc9 * ((noraab9 * ab9) - (norabc9 * bc9) + 2*(noraca9 * ca9)) +paa9 * ((2* noraaa9 *
aa9) + (norabb9 * bb9) - (noracc9 * cc9)) +pab9 * ((-1* noraac9 * ac9) + (2* noraba9 * ba9) + (noracb9 *
cb9));
prm deca10 = 1 +pabc10 * ((noraab10 * ab10) - (norabc10 * bc10) + 2*(noraca10 * ca10)) +paa10 * ((2*
noraaa10 * aa10) + (norabb10 * bb10) - (noracc10 * cc10)) +pab10 * ((-1* noraac10 * ac10) + (2* noraba10 *
ba10) + (noracb10 * cb10));
prm deca11 = 1 +pabc11 * ((noraab11 * ab11) - (norabc11 * bc11) + 2*(noraca11 * ca11)) +paa11 * ((2*
noraaa11 * aa11) + (norabb11 * bb11) - (noracc11 * cc11)) +pab11 * ((-1* noraac11 * ac11) + (2* noraba11 *
ba11) + (noracb11 * cb11));
prm deca12 = 1 +pabc12 * ((noraab12 * ab12) - (norabc12 * bc12) + 2*(noraca12 * ca12)) +paa12 * ((2*
noraaa12 * aa12) + (norabb12 * bb12) - (noracc12 * cc12)) +pab12 * ((-1* noraac12 * ac12) + (2* noraba12 *
ba12) + (noracb12 * cb12));
prm deca13 = 1 +pabc13 * ((noraab13 * ab13) - (norabc13 * bc13) + 2*(noraca13 * ca13)) +paa13 * ((2*
noraaa13 * aa13) + (norabb13 * bb13) - (noracc13 * cc13)) +pab13 * ((-1* noraac13 * ac13) + (2* noraba13 *
ba13) + (noracb13 * cb13));

```



```

prm deca34 = 1 +pabc34 * ((noraab34 * ab34) - (norabc34 * bc34) + 2*(noraca34 * ca34)) +paa34 * ((2*
noraaa34 * aa34) + (norabb34 * bb34) - (noracc34 * cc34)) +pab34 * ((-1* noraac34 * ac34) + (2* noraba34 *
ba34) + (noracb34 * cb34));
prm deca35 = 1 +pabc35 * ((noraab35 * ab35) - (norabc35 * bc35) + 2*(noraca35 * ca35)) +paa35 * ((2*
noraaa35 * aa35) + (norabb35 * bb35) - (noracc35 * cc35)) +pab35 * ((-1* noraac35 * ac35) + (2* noraba35 *
ba35) + (noracb35 * cb35));
prm deca36 = 1 +pabc36 * ((noraab36 * ab36) - (norabc36 * bc36) + 2*(noraca36 * ca36)) +paa36 * ((2*
noraaa36 * aa36) + (norabb36 * bb36) - (noracc36 * cc36)) +pab36 * ((-1* noraac36 * ac36) + (2* noraba36 *
ba36) + (noracb36 * cb36));
'-----

```

'decision parameter decb

```

prm decb2 = 1 +pabc2 *((-1* norbbc2 * bc2) - (norbab2 * ab2)) +paa2 * ((-1* norbbb2 * bb2) - (norbcc2 *
cc2)) +pab2 * ((-1* norbac2 * ac2) - (norbc2 * cb2));
prm decb3 = 1 +pabc3 *((-1* norbbc3 * bc3) - (norbab3 * ab3)) +paa3 * ((-1* norbbb3 * bb3) - (norbcc3 *
cc3)) +pab3 * ((-1* norbac3 * ac3) - (norbc3 * cb3));
prm decb4 = 1 +pabc4 *((-1* norbbc4 * bc4) - (norbab4 * ab4)) +paa4 * ((-1* norbbb4 * bb4) - (norbcc4 *
cc4)) +pab4 * ((-1* norbac4 * ac4) - (norbc4 * cb4));
prm decb5 = 1 +pabc5 *((-1* norbbc5 * bc5) - (norbab5 * ab5)) +paa5 * ((-1* norbbb5 * bb5) - (norbcc5 *
cc5)) +pab5 * ((-1* norbac5 * ac5) - (norbc5 * cb5));
prm decb6 = 1 +pabc6 *((-1* norbbc6 * bc6) - (norbab6 * ab6)) +paa6 * ((-1* norbbb6 * bb6) - (norbcc6 *
cc6)) +pab6 * ((-1* norbac6 * ac6) - (norbc6 * cb6));
prm decb7 = 1 +pabc7 *((-1* norbbc7 * bc7) - (norbab7 * ab7)) +paa7 * ((-1* norbbb7 * bb7) - (norbcc7 *
cc7)) +pab7 * ((-1* norbac7 * ac7) - (norbc7 * cb7));
prm decb8 = 1 +pabc8 *((-1* norbbc8 * bc8) - (norbab8 * ab8)) +paa8 * ((-1* norbbb8 * bb8) - (norbcc8 *
cc8)) +pab8 * ((-1* norbac8 * ac8) - (norbc8 * cb8));
prm decb9 = 1 +pabc9 *((-1* norbbc9 * bc9) - (norbab9 * ab9)) +paa9 * ((-1* norbbb9 * bb9) - (norbcc9 *
cc9)) +pab9 * ((-1* norbac9 * ac9) - (norbc9 * cb9));
prm decb10 = 1 +pabc10*((-1* norbbc10 * bc10) - (norbab10 * ab10)) +paa10 * ((-1* norbbb10 * bb10) -
(norbcc10 * cc10)) +pab10 * ((-1* norbac10 * ac10) - (norbc10 * cb10));
prm decb11 = 1 +pabc11*((-1* norbbc11 * bc11) - (norbab11 * ab11)) +paa11 * ((-1* norbbb11 * bb11) -
(norbcc11 * cc11)) +pab11 * ((-1* norbac11 * ac11) - (norbc11 * cb11));
prm decb12 = 1 +pabc12*((-1* norbbc12 * bc12) - (norbab12 * ab12)) +paa12 * ((-1* norbbb12 * bb12) -
(norbcc12 * cc12)) +pab12 * ((-1* norbac12 * ac12) - (norbc12 * cb12));
prm decb13 = 1 +pabc13*((-1* norbbc13 * bc13) - (norbab13 * ab13)) +paa13 * ((-1* norbbb13 * bb13) -
(norbcc13 * cc13)) +pab13 * ((-1* norbac13 * ac13) - (norbc13 * cb13));
prm decb14 = 1 +pabc14*((-1* norbbc14 * bc14) - (norbab14 * ab14)) +paa14 * ((-1* norbbb14 * bb14) -
(norbcc14 * cc14)) +pab14 * ((-1* norbac14 * ac14) - (norbc14 * cb14));
prm decb15 = 1 +pabc15*((-1* norbbc15 * bc15) - (norbab15 * ab15)) +paa15 * ((-1* norbbb15 * bb15) -
(norbcc15 * cc15)) +pab15 * ((-1* norbac15 * ac15) - (norbc15 * cb15));
prm decb16 = 1 +pabc16*((-1* norbbc16 * bc16) - (norbab16 * ab16)) +paa16 * ((-1* norbbb16 * bb16) -
(norbcc16 * cc16)) +pab16 * ((-1* norbac16 * ac16) - (norbc16 * cb16));
prm decb17 = 1 +pabc17*((-1* norbbc17 * bc17) - (norbab17 * ab17)) +paa17 * ((-1* norbbb17 * bb17) -
(norbcc17 * cc17)) +pab17 * ((-1* norbac17 * ac17) - (norbc17 * cb17));
prm decb18 = 1 +pabc18*((-1* norbbc18 * bc18) - (norbab18 * ab18)) +paa18 * ((-1* norbbb18 * bb18) -
(norbcc18 * cc18)) +pab18 * ((-1* norbac18 * ac18) - (norbc18 * cb18));
prm decb19 = 1 +pabc19*((-1* norbbc19 * bc19) - (norbab19 * ab19)) +paa19 * ((-1* norbbb19 * bb19) -
(norbcc19 * cc19)) +pab19 * ((-1* norbac19 * ac19) - (norbc19 * cb19));
prm decb20 = 1 +pabc20*((-1* norbbc20 * bc20) - (norbab20 * ab20)) +paa20 * ((-1* norbbb20 * bb20) -
(norbcc20 * cc20)) +pab20 * ((-1* norbac20 * ac20) - (norbc20 * cb20));
prm decb21 = 1 +pabc21*((-1* norbbc21 * bc21) - (norbab21 * ab21)) +paa21 * ((-1* norbbb21 * bb21) -
(norbcc21 * cc21)) +pab21 * ((-1* norbac21 * ac21) - (norbc21 * cb21));
prm decb22 = 1 +pabc22*((-1* norbbc22 * bc22) - (norbab22 * ab22)) +paa22 * ((-1* norbbb22 * bb22) -
(norbcc22 * cc22)) +pab22 * ((-1* norbac22 * ac22) - (norbc22 * cb22));
prm decb23 = 1 +pabc23*((-1* norbbc23 * bc23) - (norbab23 * ab23)) +paa23 * ((-1* norbbb23 * bb23) -
(norbcc23 * cc23)) +pab23 * ((-1* norbac23 * ac23) - (norbc23 * cb23));
prm decb24 = 1 +pabc24*((-1* norbbc24 * bc24) - (norbab24 * ab24)) +paa24 * ((-1* norbbb24 * bb24) -
(norbcc24 * cc24)) +pab24 * ((-1* norbac24 * ac24) - (norbc24 * cb24));
prm decb25 = 1 +pabc25*((-1* norbbc25 * bc25) - (norbab25 * ab25)) +paa25 * ((-1* norbbb25 * bb25) -
(norbcc25 * cc25)) +pab25 * ((-1* norbac25 * ac25) - (norbc25 * cb25));

```

```

prm decb26 = 1 +pabc26*((-1* norbbc26 * bc26) - (norbab26 * ab26)) +paa26 * ((-1* norbbb26 * bb26) -
(norbcc26 * cc26)) +pab26 * ((-1* norbac26 * ac26) - (norbc26 * cb26));
prm decb27 = 1 +pabc27*((-1* norbbc27 * bc27) - (norbab27 * ab27)) +paa27 * ((-1* norbbb27 * bb27) -
(norbcc27 * cc27)) +pab27 * ((-1* norbac27 * ac27) - (norbc27 * cb27));
prm decb28 = 1 +pabc28*((-1* norbbc28 * bc28) - (norbab28 * ab28)) +paa28 * ((-1* norbbb28 * bb28) -
(norbcc28 * cc28)) +pab28 * ((-1* norbac28 * ac28) - (norbc28 * cb28));
prm decb29 = 1 +pabc29*((-1* norbbc29 * bc29) - (norbab29 * ab29)) +paa29 * ((-1* norbbb29 * bb29) -
(norbcc29 * cc29)) +pab29 * ((-1* norbac29 * ac29) - (norbc29 * cb29));
prm decb30 = 1 +pabc30*((-1* norbbc30 * bc30) - (norbab30 * ab30)) +paa30 * ((-1* norbbb30 * bb30) -
(norbcc30 * cc30)) +pab30 * ((-1* norbac30 * ac30) - (norbc30 * cb30));
prm decb31 = 1 +pabc31*((-1* norbbc31 * bc31) - (norbab31 * ab31)) +paa31 * ((-1* norbbb31 * bb31) -
(norbcc31 * cc31)) +pab31 * ((-1* norbac31 * ac31) - (norbc31 * cb31));
prm decb32 = 1 +pabc32*((-1* norbbc32 * bc32) - (norbab32 * ab32)) +paa32 * ((-1* norbbb32 * bb32) -
(norbcc32 * cc32)) +pab32 * ((-1* norbac32 * ac32) - (norbc32 * cb32));
prm decb33 = 1 +pabc33*((-1* norbbc33 * bc33) - (norbab33 * ab33)) +paa33 * ((-1* norbbb33 * bb33) -
(norbcc33 * cc33)) +pab33 * ((-1* norbac33 * ac33) - (norbc33 * cb33));
prm decb34 = 1 +pabc34*((-1* norbbc34 * bc34) - (norbab34 * ab34)) +paa34 * ((-1* norbbb34 * bb34) -
(norbcc34 * cc34)) +pab34 * ((-1* norbac34 * ac34) - (norbc34 * cb34));
prm decb35 = 1 +pabc35*((-1* norbbc35 * bc35) - (norbab35 * ab35)) +paa35 * ((-1* norbbb35 * bb35) -
(norbcc35 * cc35)) +pab35 * ((-1* norbac35 * ac35) - (norbc35 * cb35));
prm decb36 = 1 +pabc36*((-1* norbbc36 * bc36) - (norbab36 * ab36)) +paa36 * ((-1* norbbb36 * bb36) -
(norbcc36 * cc36)) +pab36 * ((-1* norbac36 * ac36) - (norbc36 * cb36));

```

```

#####
#####

```

'CALCULATION OF THE REGULAR STACKING VECTORS

```

prm !sx1 = 0;          prm !sy1 = 0;
prm sx2 = (kora2 * korb2 )/3; prm sy2 = sx2 * (-1);
prm sx3 = (kora3 * korb3 )/3; prm sy3 = sx3 * (-1);
prm sx4 = (kora4 * korb4 )/3; prm sy4 = sx4 * (-1);
prm sx5 = (kora5 * korb5 )/3; prm sy5 = sx5 * (-1);
prm sx6 = (kora6 * korb6 )/3; prm sy6 = sx6 * (-1);
prm sx7 = (kora7 * korb7 )/3; prm sy7 = sx7 * (-1);
prm sx8 = (kora8 * korb8 )/3; prm sy8 = sx8 * (-1);
prm sx9 = (kora9 * korb9 )/3; prm sy9 = sx9 * (-1);
prm sx10 = (kora10 * korb10)/3; prm sy10 = sx10 * (-1);
prm sx11 = (kora11 * korb11)/3; prm sy11 = sx11 * (-1);
prm sx12 = (kora12 * korb12)/3; prm sy12 = sx12 * (-1);
prm sx13 = (kora13 * korb13)/3; prm sy13 = sx13 * (-1);
prm sx14 = (kora14 * korb14)/3; prm sy14 = sx14 * (-1);
prm sx15 = (kora15 * korb15)/3; prm sy15 = sx15 * (-1);
prm sx16 = (kora16 * korb16)/3; prm sy16 = sx16 * (-1);
prm sx17 = (kora17 * korb17)/3; prm sy17 = sx17 * (-1);
prm sx18 = (kora18 * korb18)/3; prm sy18 = sx18 * (-1);
prm sx19 = (kora19 * korb19)/3; prm sy19 = sx19 * (-1);
prm sx20 = (kora20 * korb20)/3; prm sy20 = sx20 * (-1);
prm sx21 = (kora21 * korb21)/3; prm sy21 = sx21 * (-1);
prm sx22 = (kora22 * korb22)/3; prm sy22 = sx22 * (-1);
prm sx23 = (kora23 * korb23)/3; prm sy23 = sx23 * (-1);
prm sx24 = (kora24 * korb24)/3; prm sy24 = sx24 * (-1);
prm sx25 = (kora25 * korb25)/3; prm sy25 = sx25 * (-1);
prm sx26 = (kora26 * korb26)/3; prm sy26 = sx26 * (-1);
prm sx27 = (kora27 * korb27)/3; prm sy27 = sx27 * (-1);
prm sx28 = (kora28 * korb28)/3; prm sy28 = sx28 * (-1);
prm sx29 = (kora29 * korb29)/3; prm sy29 = sx29 * (-1);
prm sx30 = (kora30 * korb30)/3; prm sy30 = sx30 * (-1);
prm sx31 = (kora31 * korb31)/3; prm sy31 = sx31 * (-1);
prm sx32 = (kora32 * korb32)/3; prm sy32 = sx32 * (-1);
prm sx33 = (kora33 * korb33)/3; prm sy33 = sx33 * (-1);
prm sx34 = (kora34 * korb34)/3; prm sy34 = sx34 * (-1);

```

prn sx35 = (kora35 * korb35)/3; prn sy35 = sx35 * (-1);
prn sx36 = (kora36 * korb36)/3; prn sy36 = sx36 * (-1);

'REFINEMENT OF THE RANDOM PART

prn tsx1 0 min = -limts; max = limts; prn tsy1 0 min = -limts; max = limts;
prn tsx2 0 min = -limts; max = limts; prn tsy2 0 min = -limts; max = limts;
prn tsx3 0 min = -limts; max = limts; prn tsy3 0 min = -limts; max = limts;
prn tsx4 0 min = -limts; max = limts; prn tsy4 0 min = -limts; max = limts;
prn tsx5 0 min = -limts; max = limts; prn tsy5 0 min = -limts; max = limts;
prn tsx6 0 min = -limts; max = limts; prn tsy6 0 min = -limts; max = limts;
prn tsx7 0 min = -limts; max = limts; prn tsy7 0 min = -limts; max = limts;
prn tsx8 0 min = -limts; max = limts; prn tsy8 0 min = -limts; max = limts;
prn tsx9 0 min = -limts; max = limts; prn tsy9 0 min = -limts; max = limts;
prn tsx10 0 min = -limts; max = limts; prn tsy10 0 min = -limts; max = limts;
prn tsx11 0 min = -limts; max = limts; prn tsy11 0 min = -limts; max = limts;
prn tsx12 0 min = -limts; max = limts; prn tsy12 0 min = -limts; max = limts;
prn tsx13 0 min = -limts; max = limts; prn tsy13 0 min = -limts; max = limts;
prn tsx14 0 min = -limts; max = limts; prn tsy14 0 min = -limts; max = limts;
prn tsx15 0 min = -limts; max = limts; prn tsy15 0 min = -limts; max = limts;
prn tsx16 0 min = -limts; max = limts; prn tsy16 0 min = -limts; max = limts;
prn tsx17 0 min = -limts; max = limts; prn tsy17 0 min = -limts; max = limts;
prn tsx18 0 min = -limts; max = limts; prn tsy18 0 min = -limts; max = limts;
prn tsx19 0 min = -limts; max = limts; prn tsy19 0 min = -limts; max = limts;
prn tsx20 0 min = -limts; max = limts; prn tsy20 0 min = -limts; max = limts;
prn tsx21 0 min = -limts; max = limts; prn tsy21 0 min = -limts; max = limts;
prn tsx22 0 min = -limts; max = limts; prn tsy22 0 min = -limts; max = limts;
prn tsx23 0 min = -limts; max = limts; prn tsy23 0 min = -limts; max = limts;
prn tsx24 0 min = -limts; max = limts; prn tsy24 0 min = -limts; max = limts;
prn tsx25 0 min = -limts; max = limts; prn tsy25 0 min = -limts; max = limts;
prn tsx26 0 min = -limts; max = limts; prn tsy26 0 min = -limts; max = limts;
prn tsx27 0 min = -limts; max = limts; prn tsy27 0 min = -limts; max = limts;
prn tsx28 0 min = -limts; max = limts; prn tsy28 0 min = -limts; max = limts;
prn tsx29 0 min = -limts; max = limts; prn tsy29 0 min = -limts; max = limts;
prn tsx30 0 min = -limts; max = limts; prn tsy30 0 min = -limts; max = limts;
prn tsx31 0 min = -limts; max = limts; prn tsy31 0 min = -limts; max = limts;
prn tsx32 0 min = -limts; max = limts; prn tsy32 0 min = -limts; max = limts;
prn tsx33 0 min = -limts; max = limts; prn tsy33 0 min = -limts; max = limts;
prn tsx34 0 min = -limts; max = limts; prn tsy34 0 min = -limts; max = limts;
prn tsx35 0 min = -limts; max = limts; prn tsy35 0 min = -limts; max = limts;
prn tsx36 0 min = -limts; max = limts; prn tsy36 0 min = -limts; max = limts;

'OUTPUT: x- and y-components of the stacking vector

prn kx1 = sx1 + (dts * tsx1); : -0.07032` prn ky1 = sy1 + (dts * tsy1); : 0.11158`
prn kx2 = sx2 + (dts * tsx2); : -0.34974` prn ky2 = sy2 + (dts * tsy2); : 0.43559`
prn kx3 = sx3 + (dts * tsx3); : 0.30954` prn ky3 = sy3 + (dts * tsy3); : -0.18555`
prn kx4 = sx4 + (dts * tsx4); : -0.09092` prn ky4 = sy4 + (dts * tsy4); : 0.14732`
prn kx5 = sx5 + (dts * tsx5); : -0.13718` prn ky5 = sy5 + (dts * tsy5); : 0.15000`
prn kx6 = sx6 + (dts * tsx6); : -0.48333` prn ky6 = sy6 + (dts * tsy6); : 0.48333`
prn kx7 = sx7 + (dts * tsx7); : 0.18333` prn ky7 = sy7 + (dts * tsy7); : -0.19475`
prn kx8 = sx8 + (dts * tsx8); : -0.15000` prn ky8 = sy8 + (dts * tsy8); : 0.12712`
prn kx9 = sx9 + (dts * tsx9); : -0.15000` prn ky9 = sy9 + (dts * tsy9); : 0.14999`
prn kx10 = sx10 + (dts * tsx10); : -0.18333` prn ky10 = sy10 + (dts * tsy10); : 0.18333`
prn kx11 = sx11 + (dts * tsx11); : -0.18333` prn ky11 = sy11 + (dts * tsy11); : 0.18333`
prn kx12 = sx12 + (dts * tsx12); : 0.15000` prn ky12 = sy12 + (dts * tsy12); : -0.15000`
prn kx13 = sx13 + (dts * tsx13); : -0.18333` prn ky13 = sy13 + (dts * tsy13); : 0.18333`

```

prm kx14 = sx14 + (dts * tsx14); : 0.48333` prm ky14 = sy14 + (dts * tsy14); : -0.47104`
prm kx15 = sx15 + (dts * tsx15); : 0.15000` prm ky15 = sy15 + (dts * tsy15); : -0.13591`
prm kx16 = sx16 + (dts * tsx16); : -0.18333` prm ky16 = sy16 + (dts * tsy16); : 0.18333`
prm kx17 = sx17 + (dts * tsx17); : -0.48333` prm ky17 = sy17 + (dts * tsy17); : 0.48333`
prm kx18 = sx18 + (dts * tsx18); : 0.18333` prm ky18 = sy18 + (dts * tsy18); : -0.18333`
prm kx19 = sx19 + (dts * tsx19); : -0.18333` prm ky19 = sy19 + (dts * tsy19); : 0.18333`
prm kx20 = sx20 + (dts * tsx20); : -0.48333` prm ky20 = sy20 + (dts * tsy20); : 0.48333`
prm kx21 = sx21 + (dts * tsx21); : 0.18333` prm ky21 = sy21 + (dts * tsy21); : -0.18333`
prm kx22 = sx22 + (dts * tsx22); : -0.15000` prm ky22 = sy22 + (dts * tsy22); : 0.15000`
prm kx23 = sx23 + (dts * tsx23); : 0.46551` prm ky23 = sy23 + (dts * tsy23); : -0.48228`
prm kx24 = sx24 + (dts * tsx24); : 0.12275` prm ky24 = sy24 + (dts * tsy24); : -0.12898`
prm kx25 = sx25 + (dts * tsx25); : -0.20678` prm ky25 = sy25 + (dts * tsy25); : 0.20385`
prm kx26 = sx26 + (dts * tsx26); : 0.42795` prm ky26 = sy26 + (dts * tsy26); : -0.43946`
prm kx27 = sx27 + (dts * tsx27); : 0.08686` prm ky27 = sy27 + (dts * tsy27); : -0.11490`
prm kx28 = sx28 + (dts * tsx28); : -0.23579` prm ky28 = sy28 + (dts * tsy28); : 0.20644`
prm kx29 = sx29 + (dts * tsx29); : 0.40895` prm ky29 = sy29 + (dts * tsy29); : -0.45536`
prm kx30 = sx30 + (dts * tsx30); : 0.08018` prm ky30 = sy30 + (dts * tsy30); : -0.13852`
prm kx31 = sx31 + (dts * tsx31); : -0.25689` prm ky31 = sy31 + (dts * tsy31); : 0.18333`
prm kx32 = sx32 + (dts * tsx32); : 0.32815` prm ky32 = sy32 + (dts * tsy32); : -0.48333`
prm kx33 = sx33 + (dts * tsx33); : -0.15000` prm ky33 = sy33 + (dts * tsy33); : 0.14992`
prm kx34 = sx34 + (dts * tsx34); : -0.15000` prm ky34 = sy34 + (dts * tsy34); : 0.15000`
prm kx35 = sx35 + (dts * tsx35); : -0.18333` prm ky35 = sy35 + (dts * tsy35); : 0.18333`
prm kx36 = sx36 + (dts * tsx36); : 0.48333` prm ky36 = sy36 + (dts * tsy36); : -0.30350`

```

```

#####
#####
#####
#####

```

'STURCTURE SECTION: FILL IN YOUR STRUCTURAL DATA

/* add kx1-kx36 and ky1-ky36 and z = i/36 from i = 0 to i = 35 to the atomic coordinates of 36 predefined rigid body like layers!

*/