

- Test Equation $T \oplus L \oplus F = 0$
- Smart Data Structure
- Test as address simulation
- Recoding H-matrix
- Deductive D-matrix
- Faults testing map
- Fault-as-address simulation
- Modeling structure for simulation
- Wire set as address modeling

Хмарний сервіс MOSI – Modeling for Simulation

Хаханов Іван, доктор філософії, асистент кафедри АПОТ ХНУРЕ – speaker,

Кулак Георгій, асистент кафедри АПОТ ХНУРЕ;

Воронов Андрій, аспірант кафедри АПОТ ХНУРЕ;

Демченко Олег, аспірант кафедри АПОТ ХНУРЕ;

Максимова Наталія, аспірант кафедри АПОТ ХНУРЕ;

Пономарьова Вероніка, інженер кафедри АПОТ ХНУРЕ;

Науковий керівник – Хаханов Володимир, д-р техн. наук, професор кафедри АПОТ ХНУРЕ



Industry AI.5.0 for Sustainable Development Goals

Industry 1.0
18th century



Industry 2.0
19-20 centuries



Industry 3.0
mid-20 century



Industry 4.0
early 21 century



Industry 5.0
21 century



Steam engine
mass production,
mechanization

Assembly lines,
mass production

Automation
IT
systems,
electronics

Cyber-physical
systems, cloud
computing, and
the IoT

Human-robot
collaboration,
mass
customization

8 DECENT WORK AND
ECONOMIC GROWTH



9 INDUSTRY, INNOVATION
AND INFRASTRUCTURE



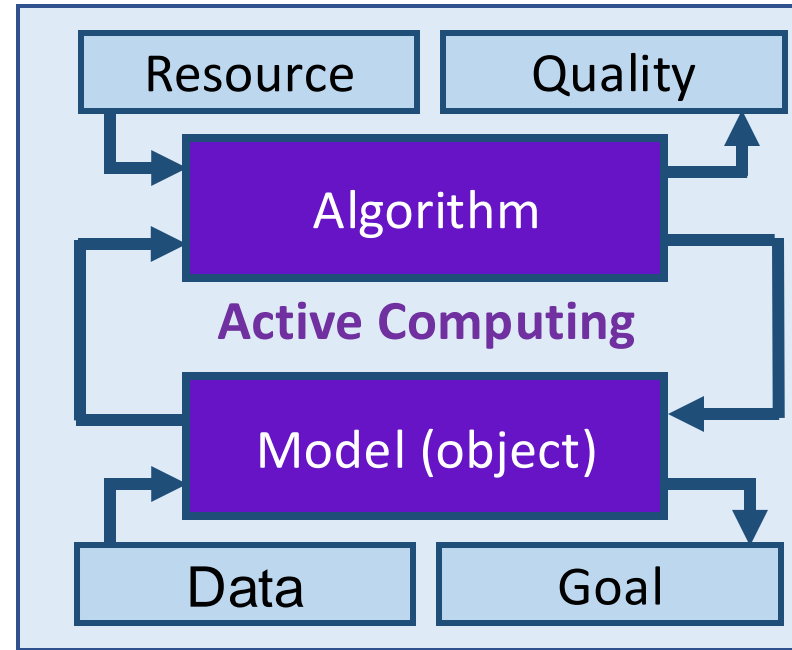
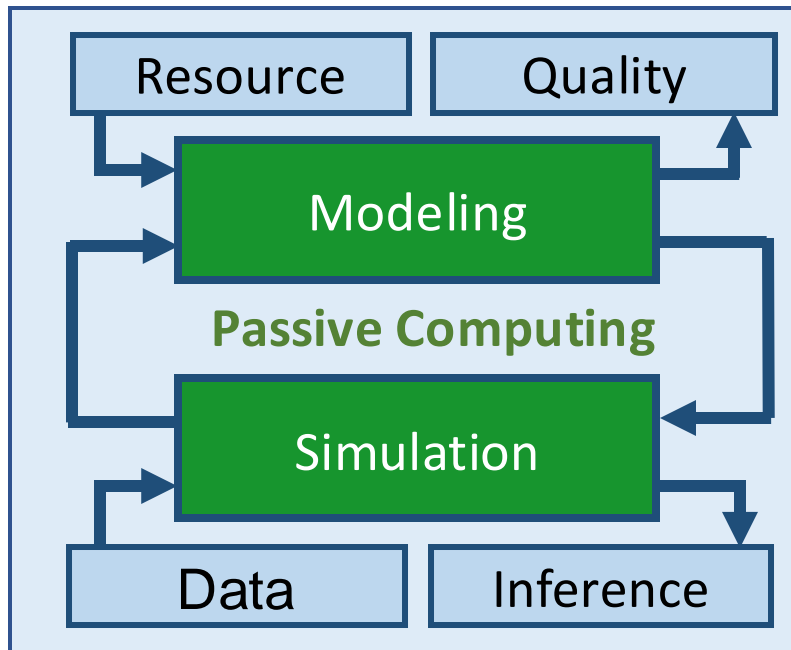
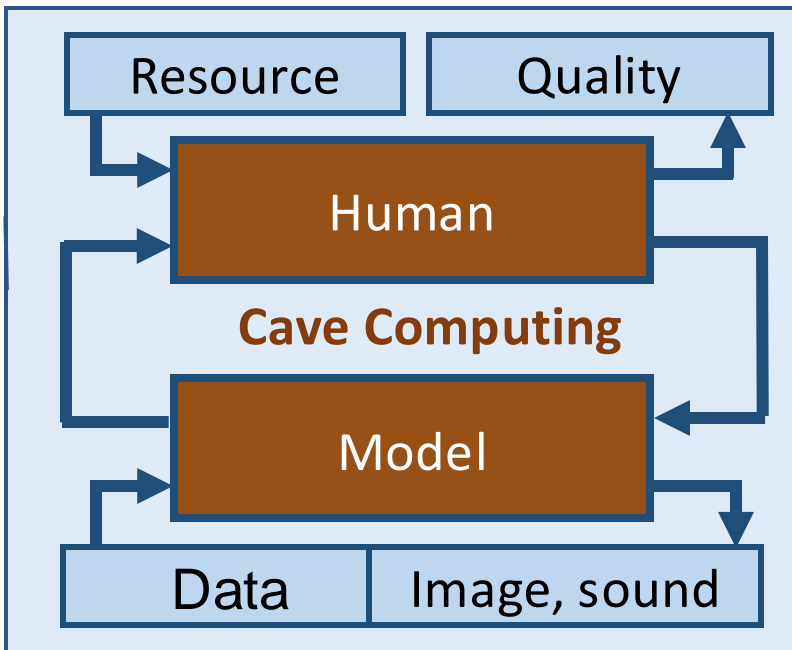
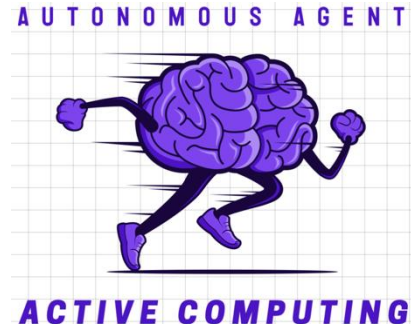
12 RESPONSIBLE
CONSUMPTION
AND PRODUCTION



Industrial Revolution Metric 5.0 поєднує в собі сильні сторони творчих людей, адаптивність, здатність вирішувати проблеми та емоційний інтелект з точністю та ефективністю комп'ютерних систем та робототехніки. Цей підхід визнає обмеженість чистої автоматизації та штучного інтелекту, особливо в складних процесах прийняття рішень, інноваціях та контекстах, які вимагають емпатії та людського судження. Індустрія 5.0 сприяє інтеграції працівників у виробничі процеси разом із передовими технологіями штучного інтелекту для підвищення продуктивності, якості продукції та загальної конкурентоспроможності. Двигуни штучного інтелекту передбачають майбутнє, в якому люди та машини працюють разом у гармонії, кожен вносить свій унікальний внесок у більш гнучкий, стійкий і чутливий промисловий і навчальний ландшафт. [Adel, A. The Convergence of Intelligent Tutoring, Robotics, and IoT in Smart Education for the Transition from Industry 4.0 to 5.0. Smart Cities 2024, 7, 325-369. <https://doi.org/10.3390/smartcities7010014>]

Підвищення продуктивності, якості продукції та загальної конкурентоспроможності безпосередньо чи опосередковано сприяють досягненню кількох Цілей сталого розвитку ООН (ЦУР). Ось найбільш значущі з них: ЦСР 8 – Гідна праця та економічне зростання; ЦСР 9 – Індустріалізація, інновації та інфраструктура.

Підвищення конкурентоспроможності потребує впровадження інновацій, модернізації інфраструктури та розвитку сталої індустріалізації; ЦСР 12 – Відповідальне споживання та виробництво. Більш ефективне використання ресурсів, зниження відходів та зменшення негативного впливу на довкілля (збереження екології планети).



99 %

0,5 %

0,5 %

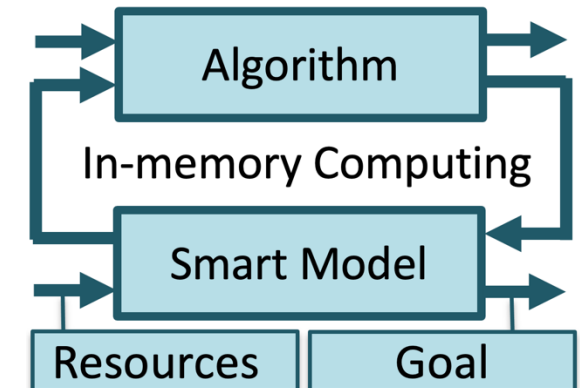
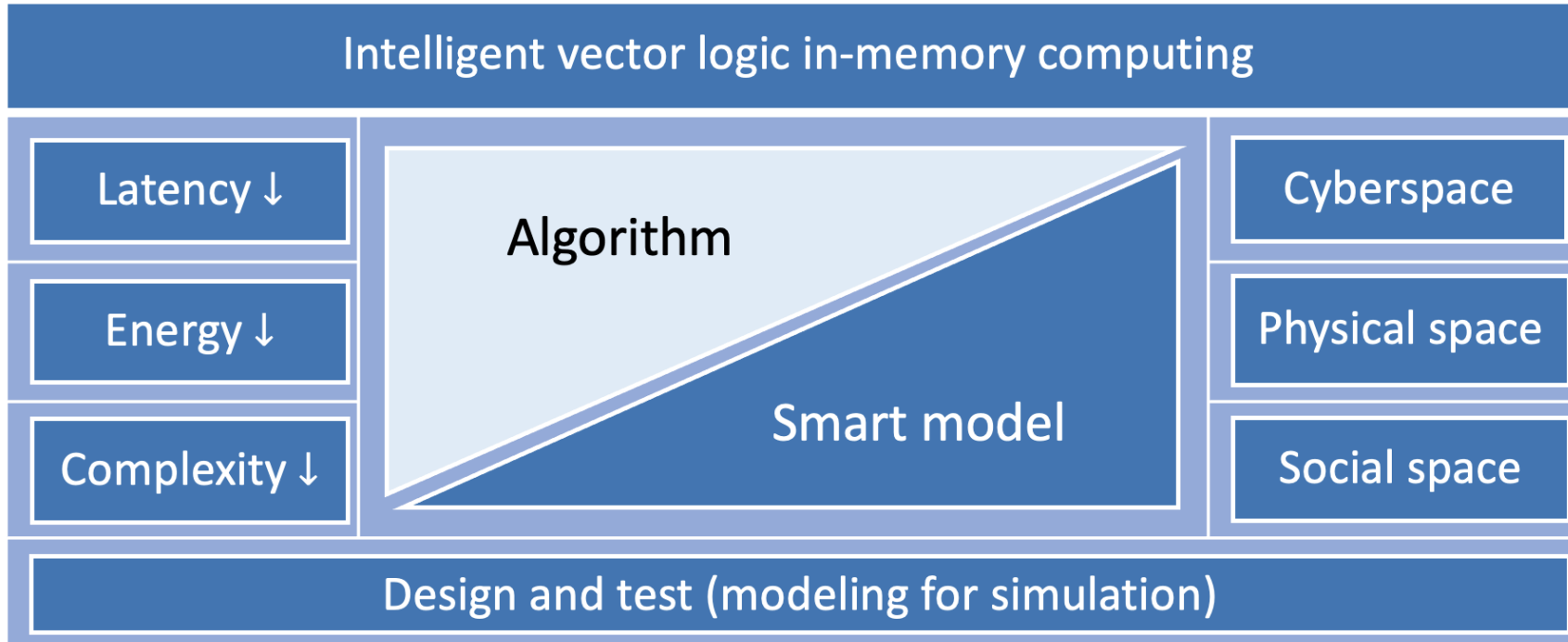
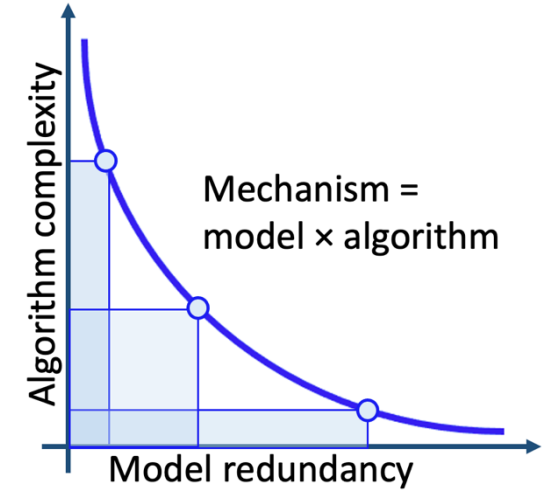
Існує всього три види обчислень (рисунок) в просторі і часі відповідно до кінцевого результату (умовиводу):

- 1) Печерні обчислення – це відображення інформації на будь-якому носії для прийняття рішень людиною.
- 2) Пасивні (імітаційні) обчислення – це моделювання вхідних впливів на моделі процесу або об'єкта з метою отримання умовиводу (прогнозування, діагностика, розпізнавання, кластеризація, класифікація).
- 3) Активні (істинні) обчислення – це виконавчі сигнали для автоматичного управління процесом або об'єктом на основі моніторингу



Intelligent vector logic computing

Комп'ютинг векторної логіки як економічно ефективний механізм інтелектуальних обчислень у пам'яті використовує транзакції читання-запису для вирішення практичних проблем аналізу та управління фізичними, соціальними та бізнес-процесами на основі моніторингу. Поняття механізму вводиться як гармонійне співвідношення між моделлю і алгоритмом вирішення задач пасивного і активного (керуючого) комп'ютингу. Для будь-якої ситуації може бути побудована така надлишкова модель, яка обнуляє алгоритм її аналізу для отримання рішення. Мета дослідження – значно скоротити часові та енергетичні витрати на управління процесами та явищами у фізичному, соціальному та цифровому світі. Засобом досягнення цієї мети є використання векторно-логічних обчислювальних механізмів у пам'яті, які значно спрощують алгоритми за рахунок експоненціальної надмірності структур даних. Розглянуто механізми обнулення обчислювальної складності алгоритму за рахунок експоненціальної складності структур даних на основі векторної логіки.



Vector-logical circuitry

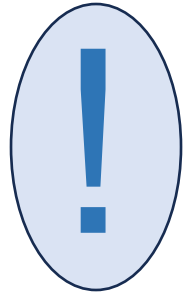
Truth Table	<table border="1"> <thead> <tr><th>X₁</th><th>X₂</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	X ₁	X ₂	Y	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <thead> <tr><th>X₁</th><th>X₂</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	X ₁	X ₂	Y	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <thead> <tr><th>X₁</th><th>X₂</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	X ₁	X ₂	Y	0	0	0	0	1	1	1	0	1	1	1	0	<table border="1"> <thead> <tr><th>X₁</th><th>X₂</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	X ₁	X ₂	Y	0	0	1	0	1	1	1	0	1	1	1	0	<table border="1"> <thead> <tr><th>X₁</th><th>X₂</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	X ₁	X ₂	Y	0	0	1	0	1	0	1	0	0	1	1	0	<table border="1"> <thead> <tr><th>X₁</th><th>X₂</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	X ₁	X ₂	Y	0	0	1	0	1	0	1	0	0	1	1	1
X ₁	X ₂	Y																																																																																														
0	0	0																																																																																														
0	1	0																																																																																														
1	0	0																																																																																														
1	1	1																																																																																														
X ₁	X ₂	Y																																																																																														
0	0	0																																																																																														
0	1	1																																																																																														
1	0	1																																																																																														
1	1	1																																																																																														
X ₁	X ₂	Y																																																																																														
0	0	0																																																																																														
0	1	1																																																																																														
1	0	1																																																																																														
1	1	0																																																																																														
X ₁	X ₂	Y																																																																																														
0	0	1																																																																																														
0	1	1																																																																																														
1	0	1																																																																																														
1	1	0																																																																																														
X ₁	X ₂	Y																																																																																														
0	0	1																																																																																														
0	1	0																																																																																														
1	0	0																																																																																														
1	1	0																																																																																														
X ₁	X ₂	Y																																																																																														
0	0	1																																																																																														
0	1	0																																																																																														
1	0	0																																																																																														
1	1	1																																																																																														
Boolean function	$Y=X_1X_2$	$Y=X_1 \vee X_2$	$Y=X_1 \oplus X_2$	$Y=\overline{X_1X_2}$	$Y=\overline{X_1} \vee \overline{X_2}$	$Y=\overline{X_1} \oplus \overline{X_2}$																																																																																										
Logical vector	Y=0001	Y=0111	Y=0110	Y=1110	Y=1000	Y=1001																																																																																										
Vector element																																																																																																
IEEE element																																																																																																
Standard element																																																																																																

Наведено деякі елементи, які мають дві змінні. Примітно, що найбільш технологічно просунута і проста форма є векторно-логічною формою будь-якого функціоналу.

Two Testing Map Synthesis Mechanisms

$V_{ijt} = \bar{A}_{it}^T \leftarrow T_{ijt}=1$ – Using the F-matrix as logic activity addresses

L-matrix: $L_i=Y_i \oplus Y$		F-matrix		T-matrix Standard		T-matrix Standard $T_{ij}=A_j^F \leftarrow T_{ij}=1$				$V_{ijt} = \bar{A}_{it}^T \leftarrow T_{ijt}=1$																			
Y	0 0 0 1 1 0 1 1	F=A \oplus A		T=L $_F$		A F 000 001 010 011 100 101 110 111				A F 000 001 010 011 100 101 110 111				A T															
0		0	1	2	3	4	5	6	7				.11	1..		11.	111				.11	1..		11.	111	000			
0		1	0	3	2	5	4	7	6		1					1.1	1.1	111			.1.			1.0	11.	110	001		
0		2	3	0	1	6	7	4	5	1											.1			1..	1.1	10.	010		
1	1 1 1 1	3	2	1	0	7	6	5	4	1	1	1				1.1					.1	.1.	.11			10.	011		
1	1 1 1 1	4	5	6	7	0	1	2	3	1						1.1					.1			0..	0.1	01.	100		
0		5	4	7	6	1	0	3	2	1	1	1				1					.1	.1.	.11			01.	101		
1	1 1 1 1	6	7	4	5	2	3	0	1												.11	1..		1.1	111	00.	001	110	
1	1 1 1 1	7	6	5	4	3	2	1	0		1										.1.			1.1	1.1	111	00.	000	111



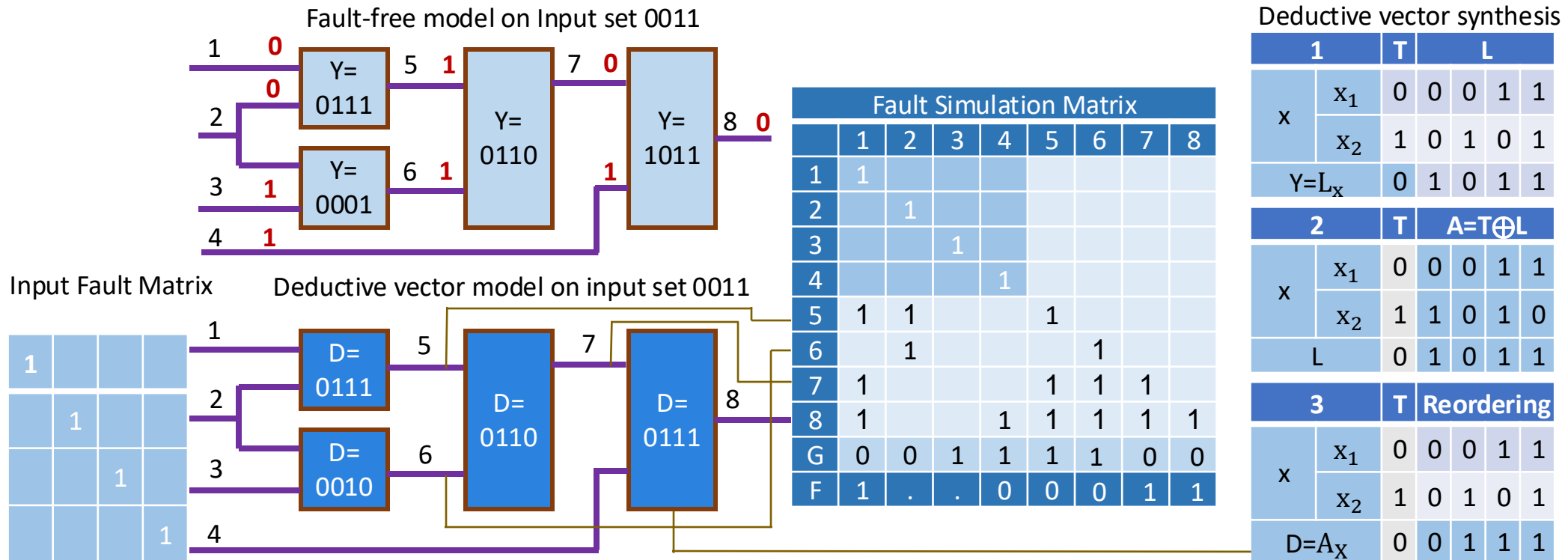
$W_{ijt} = (A_{jt}^F \text{ or } \bar{A}_{it}^T) \leftarrow T_{ijt} = 1$ – Using the F-matrix as fault codes

L-matrix:		H-matrix		E-matrix		B-matrix				$C_{ijt} = F_{jt} \text{ or } \bar{T}_{jt} \leftarrow B_{ijt} = 1$																		
Y	0 0 0 1 1 0 1 1	E=L \times F		B $_{ij}=(E_{ij})_2 \leftarrow E_{ij} \neq 0$				F 000 001 010 011 100 101 110 111				T																
0		0	1	2	3	4	5	6	7				.11	1..		11.	111				.11	1..		11.	111	000		
0		1	0	3	2	5	4	7	6		2	5				111	11			.1.	1.0		110	11.	001			
0		2	3	0	1	6	7	4	5		1	6				1..	1.1			.1	10.		1..	1.1	010			
1	1 1 1 1	3	2	1	0	7	6	5	4	3	2	1				.11	.1.	.1.			.00	.0.	.0		10.		011	
1	1 1 1 1	4	5	6	7	0	1	2	3	4	5	6				1..	1.1	1..			0..	0.1	01.		.1		100	
0		5	4	7	6	1	0	3	2			6	1											01.	.0	.10	.1	101
1	1 1 1 1	6	7	4	5	2	3	0	1	6	7	4				1..	111	1..			00.	001	0..		.01		110	
1	1 1 1 1	7	6	5	4	3	2	1	0	7	6	5				111	11.	1.1			000	00.	0.0		.0		111	

Запропоновано механізми моделювання карт тестування для вирішення задач з імітацією несправності як адрес в пам'яті без алгоритму моделювання. Карта тестування – це явна залежність між вичерпним тестом і всіма комбінаціями виявлених несправностей. Складність отримання карти тестування є константою від чотирьох матричних операцій.



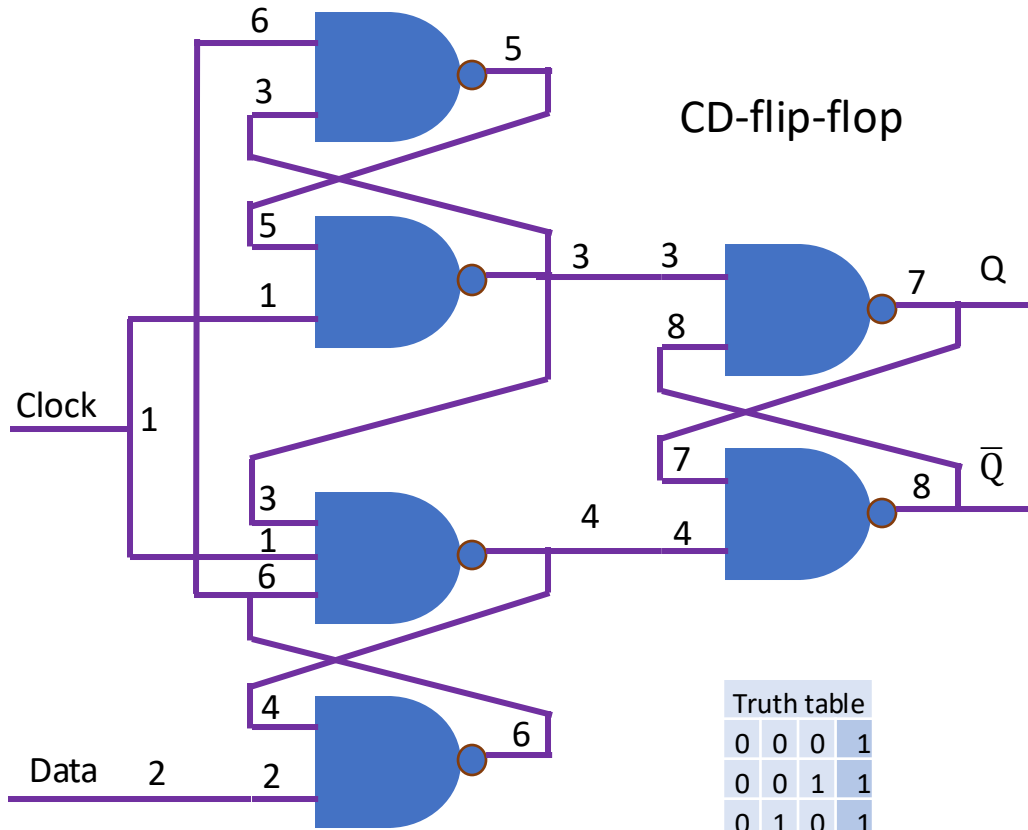
Faults-as-address circuit simulation (FAAS)



Новою метрикою механізму FAAS є механізм заповнення адреси матриці моделювання несправностей за допомогою дедуктивних векторів, які імітують комбінації несправностей у вигляді адрес. Відношення $T \oplus L = D$ між тестовою множиною T та таблицею істинності L елементу формує дедуктивний вектор для моделювання помилок, таких як адреси таблиці істинності або координати логічного вектора. З іншого боку, адреси використовуються в матриці моделювання для позначення тих n -комбінацій вхідних несправностей, які перевіряються на виході елемента. Фактично, на дедуктивному векторі елемента стовпці розглядаються як адреси n -рядків матриці моделювання для генерації одного вихідного рядка компонента. Перенесення вхідних несправностей на вихід елемента відсутнє. Тільки одиниці заповнюють рядки матриці схематичного моделювання. Матриця схематичного моделювання спочатку заповнюється одиницями по головній діагоналі. Тестовані на тестовому комплекті схем лінії замикання визначаються зворотними значеннями відмінного стану тих ліній, які мають в рядку матриці одиничні значення, що відповідають виходу останнього елемента схеми. Дедуктивний вектор виходить за рахунок співвідношень хог між тестовою множиною та логічним вектором у трьох операціях за таблицею. Перевагою запропонованого механізму FAAS є передбачувана складність алгоритму та витрати пам'яті на зберігання структур даних при моделюванні тестового набору, що визначається за формулою: $Q = N^2 + \sum_{i=1}^n (Y_i + D_i)$, N , n – кількість ліній та елементів у схемі, Y , D – множина логічних та дедуктивних векторів.



Modeling and simulation circuit with feedback loops



M	1	2	3	4	5	6	7	8	Data structure																								
1	1								Interface: Inputs 1,2, outputs 7, 8																								
2		1							D-vectors				L-vectors			I-set	Y	Inputs	NI														
3			1						0	0	0	1					1	1	1	0					0	0	1	5	1	2			
4			1	1					0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1	0	1	1	3	1	6	3
5			1		1	1			0	1	1	1						1	1	1	0					1	1	1	6	3	2		
6			1	1		1			0	0	1	0						1	1	1	0					0	1	0	4	2	2		
7			1				1	1	0	1	1	1						1	1	1	0					1	1	0	3	8	2		
8			1				1	1	0	0	0	1						1	1	1	0					0	0	1	7	4	2		
V			1	1			1	1	V – Fault integration vector from circuit outputs																								
G(t-1)=7	1	1	1	0	0	1	0	1	G – Good behavior vector of the circuit lines at t-1																								
Gt=8	0	1	1	1	1	0	0	1	G – Good behavior vector of the circuit lines at t																								
Faults			0	0			1	0	F – detected faults on the input set																								

Truth table

0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

TT

0	0	1
0	1	1
1	0	1
1	1	0

D-matrix

00			1
01		1	
10	1		
11	1	1	1

	Q	I	1	2	3	4	5	6	7	8
0	00	0.25	0.25	1		0		0	1	
1	10	0.25	0.50		0	1		1	0	
2	01	0.25	0.50	1	0			1	0	
3	11	0.44	0.81	0	0	1	0	1	0	0
4	00	0.19	0.81			0		0	1	
5	10	0.25	0.81			0	1		1	0
6	11	0.25	0.81			0	1		1	0
7	01	0.25	0.81	1	0				1	0

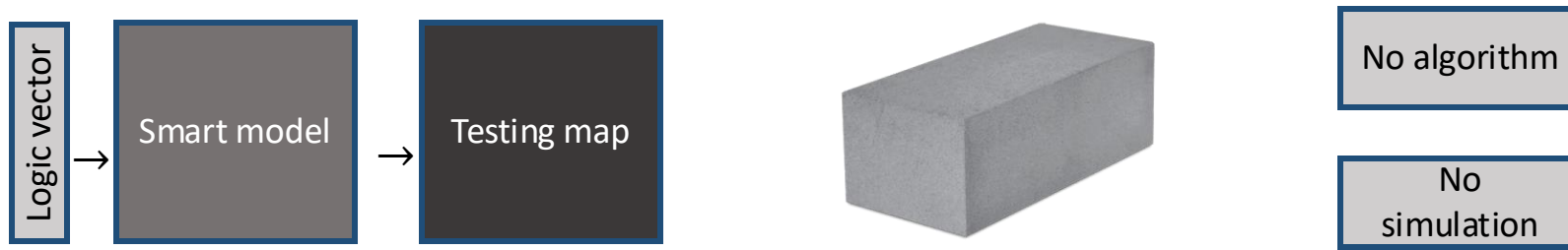
No	Fault simulation								Fault-free simulation											
	T	Q	I	1	2	3	4	5	6	7	8	M	1	2	3	4	5	6	7	8
0		0.0	0.0									0	x	x	x	x	x	x	x	x
1	0	0	0.0	0.0								1	0	0	1	1	0	1	x	x
2	1	0	0.30	0.30	0	1	0	1				2	1	0	1	0	0	1	0	1
3	0	1	0.30	0.42	1			0	1	1	0	3	0	1	1	1	1	0	0	1
4	1	1	0.40	0.62	0	0			0	1	0	4	1	1	0	1	1	0	1	0
5	0	0	0.50	0.72	1	1	0		1	0	0	1	5	0	0	1	1	0	1	1
6	1	0	0.40	0.72		0	1	1	0		0	6	1	0	1	0	0	1	0	1
7	1	1	0.30	0.82		0	0		1	0	1	0	7	1	1	1	0	0	1	0
8	0	1	0.20	0.82		0	0		1	0		8	0	1	1	1	1	0	0	1

I-set – це підготовлені вхідні набори з цього та попереднього циклу для моделювання несправностей, Y - вихідне значення при подачі підготовленого вхідного вектора.

Додано: 1) Vector $G_{(i-1)}$ попередній стан схематичної лінії. 2) M ітераційна матриця справного моделювання лінії ланцюга. 3) Символ x невизначеного стану $x=\{0,1\}$ ліній ланцюга. 4) Обмежене число ітерацій, після яких фіксуються невизначені стани до зміни рядків в останніх двох ітераціях. 5) Вектор V_x виявлення невизначеностей діаграмної лінії за 1-координатними значеннями. 6) I-set, Y – стан вводу-виводу всіх елементів схеми при відсутності несправностей на двох сусідніх циклах.



Vector-logic prompt-engineering of IP-core SoC design and test



Y		Test Map																		
0		0000			..1.	..11	.1..									1.11	11..	11.1		
0		0001			..1.	..10		.1.0							1.1.		11..	11.0		
1		0010			..0.	..01	.1..	.1.1		.101	1...				1.0.	1.01	11..	11.1		
1		0011			..0.	..00	.1..	.1.0	.10.					1.0	1.0.	1.00	11..	11.0		
1		0100		...1	..1.	..11	.0..	.0.1							1.1.	1.11	10..	10.1	101.	
0		0101		...0					.01.	.010	1...	1.0								101.
0		0110			..0.		.0..	.0.1							1.0.	1.01		10.1		
0		0111					..00	.0..	.0.0						1.0.	1.00	10..			
0	→	1000				..11	.1..	.1.1							0.1.	0.11	01..			
0		1001			..1.		.1..	.1.0							0.1.	0.10		01.0		
0		1010		...1					.10.	.101	0...	0.1								010.
1		1011		...0	..0.	..00	.1..	.1.0							0.0.	0.00	01..	01.0	010.	
1		1100			..1.	..11	.0..	.0.1	.01.				0.1	0.1.	0.11	00..	00.1			
1		1101			..1.	..10	.0..	.0.0		.010	0...				0.1.	0.10	00..	00.0		
0		1110			..0.	..01		.0.1							0.0.		00..	00.1		
0		1111			..0.	..00	.0..									0.00	00..	00.0		

Представлено розумний механізм моделювання тестових карт за структурою або функціональністю за допомогою логічного вектора: <https://vector-simulation.vercel.app/> . Алгоритму моделювання немає. Це оперативна інженерія. Редакційні колеги журналів IEEE дають негативні відгуки, не читаючи статей з третіх країн. Практика – критерій істини. Ринок університетської освіти, що налічує 25 000 університетів, чекає на ефективні, прості системи моделювання та імітаційного підказування.



Fault simulation techniques. Metrics Comparison

Fault simulation technique	Complexity	Memory	Data structure	Level	Delay	Speed	Fault model	Multy-valued
Serial fault simulation (1963)	$n \times n^3$	Predictable	Add fault model, fault list	Gate, system	No problem	Slowest	Any	Easy
Parallel fault simulation (1965)	$\frac{1}{w} \times n^3$	Predictable	Register Memory	Gate	Not capable	Middle	Logic	Difficult
Deductive fault simulation (1972)	n^2	Unpredictable	Deductive formulas	Gate	Not capable	Middle	Any	Difficult
Concurrent fault simulation (1974)	$\frac{1}{3} \times n^2$	Unpredictable	Add fault model, fault list	Gate, RTL	Capable	Faster	Logic	Easy
PPSFP – Parallel pattern single fault propagation (1985)	$\frac{1}{w} \times n^3$	Unpredictable	Add fault model, fault list	Gate	Capable	Middle	Logic	Easy
Differential fault simulation (1989)	$\frac{1}{2} \times n^2$	Unpredictable	Add fault model, fault list	Gate, RTL	Not capable	Middle	Any	Difficult
Vector fault simulation (2023)	$\frac{1}{2} \times \frac{1}{3} \times n^2$	Predictable	No, As true-value simulation	Gate, RTL, System	Not capable	Faster	Logic	Capable

Моделювання несправностей має важливе значення для АТПГ, діагностики та класифікації несправностей. Метрики симуляції несправностей включають швидкість, пам'ять, функціональні блоки моделювання та можливість затримки, послідовні схеми та багатозначне моделювання несправностей для обробки невідомого X та високоомного Z . Симулятор несправностей зазвичай потребує, крім моделі схеми, стимулів та очікуваних реакцій (які необхідні для симуляції істинного значення): модель несправностей, список несправностей. Паралельна симуляція несправностей – це симуляція, керована подіями, яка включає відмінні та погані події разом. Змоделюйте деталі несправних ланцюгів, що відрізняються від відповідних схем. Труднощі в управлінні пам'яттю. Практично всі промислові системи моделювання несправностей мають непередбачувані розміри списків несправностей і непередбачувані розміри структур даних. У всіх шести основних моделюючих рушіях використовується процесор з високим рівнем енергоспоживання. Ulrich, E.G., "Exclusive Simulation of Activity in Digital Networks," CACM, Vol. 13, pp. 10Z-II0, February 1969.

- 1) Armstrong, D.B., "A Deductive Method for Simulating Faults in Logic Circuits," IEEE Trans. on Computers, Vol. C-21, pp. 464-471, May 1972.
- 2) Ulrich, E.G. and Baker, T., "The Concurrent Simulation of Nearly Identical Digital Networks," 10th Design Automation Workshop Proceedings, pp. 145-150, 1973.
- 3) Ulrich and T. Baker, "The Concurrent Simulation of Nearly Identical Digital Networks", Design Automation Workshop Proceedings, pp. 145-150, and IEEE Computer (April 1974).
- 4) Abramovici, M. Digital Systems Testing and Testable Design / M. Abramovici, M.A. Breuer, A.D. Friedman. – New York: IEEE Press, 1990. – 657 p.
- 5) Chul Young Lee and D. M. H. Walker, "PROBE: a PPSFP simulator for resistive bridging faults," 18th IEEE VLSI Test Symposium, Montreal, Quebec, Canada, 2000, pp. 105-110.
- 6) Riahi, Navabi and Lombardi, "A VPI-based combinational IP core module-based mixed level serial fault simulation and test generation methodology," 2003 Test Symposium, P. 274-277.
- 7) Wajeb Gharibi, Vladimir Hahanov, Svetlana Chumachenko, Eugenia Litvinova, Ivan Hahanov, Irina Hahanova. Vector-logic computing for faults-as-address deductive simulation// IAES International Journal of Robotics and Automation (IJRA), vol. 12, P. 274-288, no. 3 September 2023. DOI:10.11591/ijra.v12i3. P. 274-288 Scopus.

MOSI – vector logic in-memory MOdeling for Simulation test and fault as address			
Design and test service	Complexity	Limitation	Novelty
GUI logic circuit	n^2 (line number)	Screen Area	Logic vector
Good value circuit simulation	n – linear	Screen Area	Logic vector
Circuit fault simulation	2^n (exponential)	Screen Area	fault as address
Modeling circuit logic vector	n^2 (line number)	Screen Area	Cartesian logic
Modeling logic testing map	n – linear	$2^{n=9}$ (input<10)	fault as address
Logic fault detection on test	n – linear	$2^{n=9}$ (input<10)	fault as address

The services are focused on processor-free in-memory computing via read-write transactions

Service economics – low latency, low energy, low code, low time-to-market, high yield

Хмарне моделювання для сервісів симуляції (MOSI) представлено наступними компонентами: 1) Синтез тестової карти за вектором логічної функціональності. 2) Синтез логічного вектора цифрової комбінаційної схеми, що задається логічними векторами елементів. 3) Моделювання правильної поведінки логічної схеми на основі векторного представлення логічних елементів. 4) Моделювання несправностей як адрес, так і логічних схем на основі дедуктивних векторів логічних елементів. 5) Усунення несправностей у логічній функціональності за допомогою своєї тестової карти. 6) Графічний інтерфейс для введення логічної схеми, де логічні вектори представляють елементи. Всі рішення для моделювання та імітаційних механізмів запропонованої системи mosi є оригінальними <https://vector-simulation.vercel.app/>



ВИСНОВКИ

Новизна дослідження полягає у використанні векторної логіки в архітектурі обчислень in-memory комп'ютингу на основі транзакцій читання-запису, що дозволяє зменшити споживані ресурси у вигляді енергії та часу.

Практичне значення. Практично всі промислові системи моделювання несправностей мають непередбачувані розміри списків несправностей і структур даних, тому у всіх аналогічних основних моделюючих рушіях використовується процесор з високим рівнем енергоспоживання.

Система MOSI дозволяє заощадити ресурси: енергії – мінімум на 38%; часу – мінімум на порядок.

Теоретична цінність полягає у новій теорії векторно-логічного комп'ютингу, що передбачає зробити весь майбутній комп'ютинг енергозберезуваним.

Призначення. Для вирішення проблем проектування та тестування були інтегровані хмарні сервіси. Це масштабні інженерні та економічні обчислення для університетів по всьому світу.

Ступінь завершеності розробки: діючий прототип на стадії бета-тестування. Перспективи впровадження – університети країни та зарубіжжя. Розробка може бути представлена як стартап для пошуку бізнес-ангелу.

Публікації авторів, що індексуються у Scopus:

1. Hahanov V. Vector Synthesis of Fault Testing Map For Logic / V. Hahanov, W. Gharibi, S. Chumachenko, E. Litvinova // IAES International Journal of Robotics and Automation (IJRA). – 2024. – Vol. 13, No. 3. – Pp. 293-306. – DOI:[10.11591/ijra.v13i3.pp293-306](https://doi.org/10.11591/ijra.v13i3.pp293-306)
2. Hahanov V. Vector-Logical In-Memory Simulation of Faults as Truth Table Addresses / V. Hahanov, E. Litvinova, H. Hahanova, S. Chumachenko, Z. Davitadze, I. Hahanova, H. Kulak, V. Ponomarova, V. H. Abdullayev // 2024 IEEE East-West Design & Test Symposium (EWDTS), Yerevan, Armenia, 2024, pp. 1-6, doi: 10.1109/EWDTS63723.2024.10873615.
3. Hahanov V. Prompt-Testing of Logic / V. Hahanov, D. Devadze, I. Hahanov, S. Chumachenko, E. Litvinova, V. Obrizan, P. Dmytro, A. Mishchenko, N. Maksymova // 2024 IEEE East-West Design & Test Symposium (EWDTS), Yerevan, Armenia, 2024, pp. 1-5, doi: 10.1109/EWDTS63723.2024.10873774.
4. Hahanov V. Faults-as-address simulation / V. Hahanov, S. Chumachenko, E. Litvinova, I. Hahanov, V. Ponomarova, H. Khakhanova, G. Kulak // IAES International Journal of Robotics and Automation. December 2024. Vol 13, No 4. P. 452-468. DOI: <http://doi.org/10.11591/ijra.v13i4.pp452-468>.
5. Hahanov V. et al. In-Memory Fault as Address Simulation / G. Kulak, V. Hahanov et al / 2023 IEEE East-West Design & Test Symposium (EWDTS), September 22, 2023, 1–7. <https://doi.org/10.1109/ewdts59469.2023.10297038>.

