-EN_52: Results of preparation, execution and comparison of the V machines

Main document EN_52

Project author: Víctor Estrada Díaz · Einstein–VED · UNIHOLOG.

Local update: review of 18–19 October 2026. This document describes the current state of the programs and their reports, not earlier evaluations.

1. Preparation and reading rules

Coherence must be ensured: the internal state of V must be guaranteed unknown and all qbit inputs and outputs must be unknown and independent, except those defined as 0 or 1. Everything is prepared so that, from a state of not knowing, it collapses upon measurement at $t = t_0$.

First, the explicitly fixed values and their logical consequences are propagated. The relations between unknown signals are then retained without assigning outcomes to them in advance. Knowing the relation does not amount to knowing the values of a prepared execution.

In this protocol, neither Q nor V is required to read or determine its internal phase before measurement. Signal transformations are internal operations, not observer readings. Prior tests of a formula with trial values are not observations of the already prepared machine.

V execution does not receive Q results. The V reports are saved first; comparison with the Q reference is performed afterwards, using another program and only for cases with a recorded measurement.

V is implemented here using conventional deterministic software. Neither V_ref, a global amplitude register, trajectory enumeration nor a Q outcome sampler is used to produce its output.

Internal composition clarified by the author

A V machine may contain modules that are other V machines:

$$V_t(V_1, \ldots, V_m), \qquad \mathbf{x}_i(t_i) \in \{0,1\}^{r_i}.$$

Relations are prepared without knowing unfixed variables. When module inputs become fixed at t_i, those same values propagate, preserving dependencies between modules. Values already declared as 0 or 1 and their consequences propagate first.

In the single-trigger protocol at $t = t_0$, internal propagation responds to that trigger and does not introduce earlier external observations. This clarification resolves the composition criterion requested for V2; no preparation different from the author's definition is requested. Implementation and verification continue to be documented separately. V1 remains established; V2 and V3 are executed and checked separately.

Vf: independent test and precondition for V2

V construction rule: statistics are not part of the circuit. No state or output is generated using distributions, estimators, histograms or random draws. Predictions and comparisons, if performed, are external and do not condition execution. Prior lack of knowledge is not replaced with a sampler. This rule is also preserved in the project's local construction instructions.

The Vf module receives connected Q_vbit inputs and their relations, not a tuple of bits fixed by its adapter. During preparation it neither accesses their values nor evaluates signals. Vf fixes its inputs inside `colapsar(t0)`, using the same outer trigger; only then does it propagate values. It adds no clock, random draws or independent measurements. A completed output is retained without repeating collapse.

The Vf precondition has six tests of its own, independent of V2. They check preparation, blocked reads, the shared time, cached output and different module widths. The eight connection tests are reported separately. These tests are not added up as a certification of V2.

Vf receives direct or related Q_vbits and fixes them only inside its trigger. It evaluates the connected graph once from that common capture, retains every node value and applies the Fourier wiring to the resulting port values. Shared sources retain the same value: no fresh outcome is chosen in each module. Output signal pairs reach the comparator using references from the same event. This is deterministic propagation of captured values; it is not defined to be joint quantum evolution.

Before composing V2, the Vf report is checked against the current code. V2 then uses the module's input, transformation and output contract. The broader hypothesis of equivalence with Fourier Q is distinct from this verified protocol: it is not established merely because inputs are unknown before the trigger. Nor is that equivalence required as a precondition for constructing or executing V.

V2: connection and diagnosis of the composition with Vf

The composed realization connects modular relations to Vf's Q_vbits and retains the output comparator. The common trigger reaches Vf before references are consulted. It retains the installed V2 plan, with no global amplitude register and no Q results introduced into the execution.

The V results contain both complete real-clock executions and the declared profile. The profile configures deterministic references, not a statistical mechanism or execution condition. Both cases report `MEDIDO`, with captured sources, the graph, internal node values and output. `test_v2.py` now uses this realization by default; the earlier shared-dependency block is not retained. The subsequent comparison runs after outputs are saved, without feeding them back to V. Its results, including disagreements, remain in the report. Construction, execution and comparison are separate steps: `MEDIDO` does not itself establish Q2 equivalence.

2. Comparator connected to the latch

With the author's authorization, the readout from section 12.2 has been connected:

$$w = \frac{|s_0|^2}{|s_0|^2 + |s_1|^2}, \qquad r(t) = \text{frac}\left(\frac{\omega t + \eta}{2\pi}\right),$$

$$b(t_0) = \begin{cases} 0, & r(t_0) < w, \\ 1, & r(t_0) \geq w. \end{cases}$$

The temporal comparator implements this rule without generating random numbers. Two zero signals invalidate the reading; they are not assigned an arbitrary outcome. Equality with the threshold belongs to the branch that returns 1.

The software reference has a default configuration of $\omega = 2\pi$ radians per second and $\eta = 0$. It is a deterministic periodic reference. Configuration values and the reference actually captured are recorded in the report. The threshold is calculated from the declared formula: it is not estimated from frequencies and the output is not randomly drawn. The probabilistic discussion in section 12.2 concerns external predictions; it is neither a component nor an execution precondition of V.

CircuitoSenalesV prepares and executes the signal transformations without returning their values to the caller. H, phase, RX and RZ use the formulas in operaciones_senales.py. CNOT and phase connections between units are executed in cases that the component can perform without losing the joint relations. Diagonal phases at the end of the circuit are retained as branch relations, without building a joint amplitude register. This rule is not applied if a subsequent H or RX mixes the branches. A general connection that has not yet been implemented explicitly halts execution: it is not replaced by two independent units.

In V1, the oracle target retains the signal relation corresponding to the inversion eigenvalue −1. The CNOT applies that sign to the corresponding control channel, preserving the target. This is an algebraic application of the connection, not an intermediate measurement or an oracle classification delivered as the answer.

The latch triggers a joint capture. The instant t_0 comes from the monotonic clock of that execution, not from a timestamp supplied to select the outcome. The comparator fixes the source bits and those same values propagate through the final binary relations until the outputs are determined. No new reading is taken at each node, nor is an independent bit chosen for each output. Relations already fixed by the inputs are simplified beforehand.

The final relations are prepared with propagacion_conocidos.py. `resolver_captura` accepts only an immutable tuple of captured bits; it rejects a deferred source without reading it. It also rejects a capture that contradicts an explicitly fixed value. Resolution does not consult Q, consult the clock again or access the signals.

This implementation applies propagation to the final binary relations. It does not move capture across an intermediate H to avoid a pending signal coupling. The generic component prepares its signal prefix once; the deferred V3 executor also postpones that calculation until the trigger. The nodes of the prepared relations are resolved. That work is included in the measurement time and is not declared to have constant cost.

The capture retains the source bits, the propagated node values and the output bits. The audit also records the real and imaginary components of the signals in the prefix that feeds those relations, the threshold and the reference at the same logical instant. The reading and audit interfaces remain locked before triggering. Querying again returns the existing record and does not repeat the measurement. A coherent software capture is not presented as a measurement of physical hardware simultaneity.

The current component does not reuse a common reference as though it justified several independent unknown readings. If it needs more than one unfixed reading, it accepts references explicitly connected by port. It rejects presenting a single instance as several sources. The existence of distinct references does not certify their independence. Without those connections, it reports which ones are missing. That limitation belongs to this component; it is not added as a universal premise of V.

3. Current data, programs and results

Machine	Program	Inputs	Report
V1	test_v1.py	data_v1.json	resultados_v1.json
V2	test_v2.py	data_v2.json	resultados_v2.json

Machine	Program	Inputs	Report
V3	test_v3.py	data_v3.json	resultados_v3.json

All three programs use pruebas_v.py. V2 defaults to the composition with Vf and its declared readout profile. V1 retains its circuit; V3 uses the deferred executor and the same comparator. Each case prepares a new machine and requests a single measurement. The data contain the case inputs, not Q answers. The reports retain inputs, SHA-256 hashes, preparation and measurement times, the recorded instant and capture, when it completes.

V1: established result for Deutsch's task

The result is established by the exact argument in section 7 and the verifiable report, in addition to the four recorded measurements below.

Bit order: `[q0, q1]`, with q0 as the least significant bit. The Deutsch protocol comparison uses the first bit, q0.

Oracle input	Input table	q0 obtained	q0 predicted by the Q reference
Constant 0	`[0,0]`	0	0
Constant 1	`[1,1]`	0	0
Identity	`[0,1]`	1	1
Inverter	`[1,0]`	1	1

The four recorded first bits match the protocol's Q reference. The subsequent Q calculation assigns each a probability of 0.9999999999999996, equivalent to 1 within the numerical precision used. The q1 values and the full capture of each execution are in the JSON report; they may change when triggering again at another instant.

The agreement of these four first bits is not used to claim a complete joint distribution or independence between preparations. The q1 output and the statistical quality of the reference require their own checks. Nor has a time advantage over Q hardware been measured.

V2 and V3: audited rules and default executions

The rule audit records **14 successful tests** of V2 and V3. Preparation does not consult sources or fix unknown outputs. A single external clock records the trigger and every module uses that same instant. Propagation does not reread sources; later reads return the saved capture, and a failure does not authorize another attempt on the same preparation.

V2 retains its shared graph, and Vf fixes the connected Q_vbit inputs inside its own trigger. In deferred V3, preparation stores only constants and connections: signal calculation, resolution and readout run once inside the event. The V3 profile declares deterministic periodic references for all three ports. Outputs are not generated using probabilities, random draws or statistics; the comparator threshold is an already declared deterministic operation.

The tests verify the software access protocol and connections, not physical isolation against inspection or statistical independence of deterministic sources. Hiding a Python variable does not establish those properties. The Q comparison remains external and subsequent.

Machine and input	Status	Stopping point
V2, $N = 15$, base 2	`MEDIDO`	Modular composition, Vf and completed output
V2, $N = 21$, base 2	`MEDIDO`	Modular composition, Vf and completed output
V3, three units, H-CZ-RZ	`MEDIDO`	Deferred preparation, declared references and common capture; terminal phases retained

V2 and V3 record their bits and permit subsequent external comparison. V3 no longer defaults to the component without connected references. The phase test described below retains its own scope; it is not confused with the default execution or added as another global proof.

V2: review of the blocks and connection size

verificar_v2.py has been executed without consulting Q. The report is verificacion_v2.json. The original data and the declared V2 connections are preserved.

Input	Units	Controlled multiplications	Gates in the complete plan	Block captures with fixed inputs	Verified modular chains
$N = 15$, base 2	12	8	56	256	6
$N = 21$, base 2	15	10	80	640	6

These are **896 component captures and 12 multiplication chains**, not 908 measurements of complete V2. Both controls, all the small work words in these tests and the preservation of words outside the modulus were checked. These test tables are not part of V's internal operation.

The chain computes the declared relation $y \mapsto a^x y \bmod N$ without receiving a Q result.

With $b = \ bit_length(N)$, V2 prepares $3b$ units, $2b$ modular blocks and an inverse Fourier transform on $2b$ units. Its compiled plan contains exactly:

$$g(b) = 2b^2 + 6b\,.$$

That count has been checked for seven sizes, between 2 and 64 bits of the modulus. It is a quadratic bound on the number of connections, not a proof of the total execution time of an implementation that is still incomplete. The arithmetic of the modular blocks also has a cost that depends on the length of its operands.

The generic signal adapter, no longer V2's default realization, stops at the first controlled multiplication: operation index 8 for $N = 15$ and index 10 for $N = 21$, counting from 0. The report now retains the exact index, ports, multiplier and modulus, as well as the preparation time consumed. What is missing in this component is the connection of the joint signal relation of the unread control and the work register to the subsequent Fourier transform. **A latch is not missing, nor is knowing the phase before measurement required.** That coupling has not been replaced by an early measurement, a period search or a global amplitude register.

V2: modular composition with controls not yet read

verificar_puente_v2.py composes the existing modular relations and the input SWAPs of the Fourier without assigning values to the controls. The bridge report preserves the connection graph: 20 nodes for N=15 and 72 for N=21, without enumerating joint states. Test values are applied only in the component checks; they are not read from a prepared V2.

For N=15, six Fourier H gates are deduced to act on free factors and fix their output to zero; their controlled phases with a zero port have no effect. The first H requiring signals related to the work has index 47 in the complete plan. For N=21 it has index 25. Indices start at zero. These are deductions from the connections, not early measurements of the phase or the controls.

This check analyzes the signal adapter, not the execution of the new composition with Vf. Its indices and deductions are not presented as V2 outputs. Current captures are in the execution reports linked above. The deferred V3 execution has its own tests.

V3: connected final phases and test capture

verificar_v3.py has been executed. The report verificacion_v3.json retains four branches with test references, one capture with a real clock and the hashes of its original code and data.

The final CZ is not replaced by two local phases. The joint phase relation is retained, which for the protocol's captured bits is:

$$d(b_0, b_1, b_2) = \exp(i\pi b_0 b_1 + i\frac{\pi}{2}(b_2 - \frac{1}{2}))\,.$$

The justification for a terminal phase diagonal in the readout basis is:

$$|d(x)| = 1, \qquad |d(x)a_x|^2 = |a_x|^2 .$$

This algebraic equality does not require enumerating the values of x in the program. The phase and its ports remain prepared as a relation; the report retains its factor for the captured branch. **This equality is not used to erase a phase before a subsequent H or RX.** The regression test explicitly verifies that limit.

The four branches verified with test comparator inputs are `(0,0,0)`, `(0,1,0)`, `(1,0,0)` and `(1,1,0)`. It is also verified that the phase is evaluated on the propagated outputs, not on source bits that a CNOT might have changed. The angle of a terminal CP is also retained. Neither any reference nor the clock is consulted before triggering; rereadings do not produce a second capture.

The test with a real clock uses two declared periodic references:

$$\omega_0 = 2\pi, \qquad \omega_1 = 2\pi\sqrt{2}, \qquad \eta_0 = \eta_1 = 0 .$$

The specific capture, its t_0 and the reference values are in the report. This configuration is a **connection and reading test**, not a declaration of proven independence or uniformity. The configuration is not changed silently: the current default executor and its profile are identified above. A single capture is not used to declare the entire coherence preparation satisfied. Q is consulted only after the V reports have been saved.

For the family of local H gates followed exclusively by terminal CZ phases, 2, 8, 32 and 128 units were tested. Two signals per unit and one relation per phase are stored, not 2^n states. With fixed precision, preparation and resolution of this family cost $O(n + g)$ in time and memory, with g phases and their connections. The test uses identified test references; it does not demonstrate the statistical quality of a physical source or extend the bound to arbitrary circuits with interference.

4. Subsequent comparison with Q

comparar_resultados.py is executed after the three reports have been saved. It checks that each corresponds to its input file and the current code. It rejects results with outdated code hashes instead of reusing them as current. It retains the hash of the V report used. The result is in comparacion_maquinas.json.

Tally of the default executions: **7 V cases, 7 completed measurements, no pending connection and 7 subsequent comparisons**. There are also two comparisons of separate tests: H-CNOT and V3 with declared references not validated as independent. Those tests are not counted as completed default executions. The V1 comparison is made with the existing local Q reference in `DEMUESTRA/comparacion/q1_v1.py`, without modifying it or executing it within V. It is a Qiskit reference calculation, not a measurement of a physical Q machine. Its hash is retained.

The reference modules are loaded only for measured V cases. The reference cannot provide an answer to the V process, which has already finished and saved its report before this checker is executed.

Existing H-CNOT circuit: joint propagation

The existing portable program MaquinasV/par_h_cnot/v.py has also been executed, without changing its H and CNOT connections or replacing it with a detector. With both inputs prepared as 0, the final relation is `salida_1 = salida_0`: a single unfixed value determines both outputs.

The execution with a real clock saves its specific output in the report, always with both bits equal. Two prior tests with a controlled clock verified both `(0,0)` and `(1,1)`. The source and output values, the nodes and the same t_0 are recorded in resultados_propagacion_colapso.json. Repeated reading returns that capture and does not execute a new collapse.

After saving the V capture, the reference for the corresponding QASM has been calculated. The local Q reference assigns a probability of approximately 0.5 to each of those two outputs. The record of this additional comparison appears in `circuitos_adicionales` in the comparison JSON. A single capture is not presented as proof of a distribution or of a Bell experiment.

5. Local component checks

Direct execution against known results

ejecutar_tareas_conocidas_v.py executes the installed conventional V software without a Q engine or Q simulation. The external data declare known references before execution; only operational inputs reach V. These are not Q hardware measurements. There is one capture per case, without choosing instants or repeating until success.

The report records four matching V3 cases: H-H returns `[0,0,0]`, H-RZ(π)-H returns `[1,0,0]`, RX(π) on the third port returns `[0,0,1]`, and the H-H-CZ-RZ readout belongs to its four admitted outputs. The first three check a unique output; the fourth does not establish equality of distributions.

In V2, the new readouts for N=15 and N=21, base 2, are both 28 and yield no factors with the same classical postprocessing used by Q2. Only that classical function is loaded, not the Q module. A Q2 readout can also be unhelpful; two captures establish neither general difference nor equivalence. The evaluation of earlier captures is separately identified: readout 740 for N=21 does recover 3 and 7.

The V3 family with H-H per unit returns zero for 2, 8, 32 and 128 units. Its two gates and two channels per unit admit linear traversal at fixed precision; timings are in the JSON. This does not establish the order of arbitrary V3 circuits. The quadratic count of the V2 plan does not by itself

establish the total order of factoring. Q simulation timings are not used, and no Q hardware timings are available here to establish equal orders.

Component	Check performed	Report
Comparator and latch	9 comparison rules, 5 rejections of invalid inputs, 8 captures with a test clock and 1 with a real clock	resultados_comparador_temporal.json
Conventional binary gates	38 measured cases matching their truth tables, with fixed inputs	resultados_puertas_v.json
Knowledge propagation	102 patterns, 6 composite relations and 5 invalid declarations rejected	resultados_propagacion_conocidos.json
Signal relations	10 algebraic identities with variables without assigned values and 3 numerical H-H regressions	resultados_relaciones_senales.json
Propagation of a capture	2 composite cases, 32 cases of the modular relation, 2 captures with a test clock and the H-CNOT program with a real clock	resultados_propagacion_colapso.json
V2 components	896 block captures with fixed inputs, 12 chains and connection counts for 7 sizes	verificacion_v2.json
V3 terminal phases	4 test branches, CP, phase after CNOT, 4 sizes up to 128 units and one capture with a real clock and declared references	verificacion_v3.json

Component	Check performed	Report
V2/V3 rules	14 tests: preparation without reading, common trigger, propagation, single capture and cores without random draws or statistics	auditoria_reglas_v2_v3.json

These checks differ and are not added together as though they were all measurements of complete machines. Test clocks are identified as such and are not presented as executions with a physical trigger clock. Propagation checks identities such as `0 AND A = 0` and `A XOR no-A = 1` without assigning a value to A. The symbolic formulas are not presented as readings of the prepared machine.

6. Local reproduction

From `Traductores_V/Pruebas_V` :

```
python test_v1.py
python test_v2.py
python test_v3.py
python test_propagacion_colapso.py
python verificar_v2.py
python verificar_v3.py
python auditar_reglas_v2_v3.py
python comparar_resultados.py
```

Default paths are resolved from the program directory. The tests return code 0 when all measurements are completed and code 2 if any remain pending. Caught programming failures appear as `ERROR_EJECUCION`, separately from pending connections. The programs' own reports are regenerated; input files are not modified.

V execution uses the Python standard library. The Q reference is installed with requirements.txt and the optional algebraic tests with requirements_relaciones.txt.

```
python test_comparador_temporal.py
python test_puertas_v.py
python test_propagacion_conocidos.py
python test_relaciones_senales.py
python test_propagacion_colapso.py
```

If this folder is moved outside the local tree, the Q reference modules must also be provided for the subsequent comparison. V does not need them to run. Nothing has been published on the web and no Git operations have been performed.

The complete, ordered review with report regeneration is executed with ejecutar_revision_v.py. It verifies V1, reviews V2 before moving on to V3 and executes Q only at the end. The local control checks the originals, DEMUESTRA and the existing translations. The review report distinguishes successful technical tests from complete machines that are still pending. An exit code accepted for a pending item does not become certification of that pending item.

7. V1 work bound: no internal analysis of exponential states

The installed V1 has been audited using analizar_complejidad_v1.py. The verifiable report is in analisis_complejidad_v1.json, with the hashes of the code and results examined. This analysis does not execute Q or perform new V measurements.

Resources of the four executions

V1 case	Units	Plan gates	H	Oracle NOT/CNOT	Local complex signals	Comparators	Final nodes
Constant 0	2	3	3	0	4	2	3
Constant 1	2	4	3	1	4	2	3
Identity	2	4	3	1	4	2	3
Inverter	2	5	3	2	4	2	3

The four signals are two channels for each of the two units, not the entries of a global vector indexed by the states of a joint basis. The program does not enumerate configurations or trajectories to select an answer. Nor does it repeat capture until it matches a Q result. There is a single trigger per preparation.

Complexity argument

The bound does not rely solely on the absence of an exponential table. The operations performed by the code are inspected:

1. The test oracle is compiled from its two input bits into a maximum of one NOT and one CNOT. It does not receive a classification calculated by Q.
2. Preparation traverses a plan of between three and five gates. Each H, NOT or CNOT used by this V1 performs a bounded number of operations on its local signals; it does not call a joint state evaluator.
3. Triggering obtains a temporal reference, applies two comparators and resolves three final nodes. It does not wait for a favorable phase, search for an output or measure the sources again during propagation.
4. The working lists and records have sizes bounded by those units, gates and nodes. The symbolic algebra of the tests is not executed within this numerical V1 path.

With fixed precision arithmetic, the core work per execution admits a bound of the form:

$$T_{V1} \leq C_0 + 5C_{\text{puerta}} + 2C_{\text{comparador}} + 3C_{\text{nodo}},$$

and its working memory a bound of the form:

$$M_{V1} \leq D_0 + 4D_{\text{señal}} + 5D_{\text{conexion}} + 3D_{\text{nodo}}.$$

The constant terms include preparing the structures and recording a capture. They do not include the subsequent Q simulation, which is another checking process and does not produce V's answer.

Therefore, for **this two-unit V1 and the implemented Deutsch protocol**:

$$T_{V1} = O(1), \qquad M_{V1} = O(1)$$

per execution. For R repetitions of the same protocol, the work is $O(R)$; retaining all reports requires additional memory $O(R)$. The bound concerns algorithmic operations, not a guarantee of operating system latency or equality in seconds with a Q device.

Conclusion: the executed V1 does not incorporate a search or an analysis of internal states of exponential size. It produces its output through its signal operations, comparator and relations, not by simulating Q1. This result concerns the program and protocol analyzed; it is not used to claim the scaling of V2/V3 or of an extension to a variable number of units that this V1 does not implement.

To repeat the analysis after generating `resultados_v1.json`:

```
python analizar_complejidad_v1.py
```

Established V1 result

For the declared preparation and circuit, the final control pair is:

$$s_0 = \frac{(-1)^{f(0)} + (-1)^{f(1)}}{2}, \qquad s_1 = \frac{(-1)^{f(0)} - (-1)^{f(1)}}{2}.$$

If the function values agree, the second channel is zero; if they differ, the first is zero. Thus, for any readout reference admitted by the comparator, the bit solving the task satisfies:

$$q_0 = f(0) \oplus f(1) .$$

The same answer to Deutsch's problem is established for V1 and ideal Q1 for all four possible binary functions, with one capture per preparation and the bounded cost above. It does not depend on choosing a favorable instant. This conclusion concerns the bit solving the task; it does not characterize every distribution or Q hardware.

asentar_v1.py checks the argument with exact arithmetic and 16 trigger tests, and verifies the fingerprints of the four real-clock captures and their subsequent Q1 comparison. The established result preserves this evidence. V2 and V3 have their own executions and audits; their checked rules do not replace evaluation of the tasks solved.

Dual license: ethics and use

Text: **CC BY-NC-SA 4.0**. Sharing, adaptation, noncommercial use, attribution and share alike. Code: **HL3**, in accordance with the author's ethical declaration; commercial use permitted only if its conditions are met. Prohibited: wars, human rights violations, environmental harm, exploitation, mass surveillance and discrimination. Declared principles: respect for human dignity, no use for profit without permission and prohibition of harmful uses.
Author's license · CC BY-NC-SA 4.0 · HL3.