

SynCore v2.1：可重構融合一流動運算架構

v2.1 / 2026-07-30 / Neo.K（許筌崴） / 公開概念技術論文／可驗證架構提案

<https://thisoneisneok.com/html/papers/ccp-16.html>

SECTION

從可組合超純量、數位資料流到耦合振盪加速器的統一設計

英文題名：SynCore v2.1: A Reconfigurable Fusion-Flow Computing Architecture

文件編號：EML-SYNCORE-2026-v2.1

作者：Neo.K（許筌崴）

協作修訂：Aletheia

機構：一言諾科技有限公司（EveMissLab）

日期：2026年7月30日

文件性質：公開概念技術論文／可驗證架構提案

證據等級：E0（尚未完成專用晶片、FPGA 原型或可重現基準）

前版：SynCore v1.0（神核融合引擎）；SynCore v2.0（流動融合統一版）

授權：公開發布前由作者確認最終授權條款

SECTION

摘要

本文提出 SynCore v2.1，一種將可組合單執行緒資源、數位資料流加速、物理耦合振盪運算、精確狀態提交與跨層控制納入同一系統的異質運算架構。此版本不是對「多個核心可以無條件融合成數倍單核效能」的再次宣告，也不把 Kuramoto 同步視為可替代一切布林運算的宇宙原生計算。它重新界定 SynCore 的研究問題：

在功率、熱、互連、正確性、重組成本與工作負載結構的共同約束下，如何讓計算資源在獨立核心、融合叢集、數位資料流與物理流動加速之間，進行可驗證的模式選擇與協同？
SynCore v2.1 包含五類核心元件：

- Scalar Tile（純量執行晶粒）：提供相容的通用指令執行；
- Fusion Cluster（融合叢集）：以共享或分散式前端、執行視窗、暫存器與執行單元提高單執行緒的指令級與記憶體級並行；
- Digital Flow Array（數位流動陣列）：以顯式資料依賴與空間映射執行規則、張量、圖、串流和迭代核心；
- Oscillator Compute Tile（振盪器運算晶粒）：針對可映射到相位耦合、能量最小化或同步動力學的特殊問題提供近似解；
- Verification and Commit Engine（驗證與提交引擎）：負責精確狀態、檢查點、重放、錯誤隔離與結果資格判定。

新版明確區分三種互連尺度：片上 SynFabric 用於低延遲資源協同；封裝內 UCIe 類介面用於晶粒間高頻寬傳輸；CXL 類介面用於系統級一致性記憶體、設備與資源池化。CXL 不再被描述為「無限互連」，也不承擔單週期跨晶片核心融合。

本文提出 SynIR 中介表示、模式切換狀態機、多目標編排模型、五項可證偽命題，以及從模擬、FPGA、RISC-V 叢集、振盪器電路到 chiplet 原型的分階段驗證路線。SynCore 的可公開技術主張因此被縮減為：它是一個融合既有可重構微架構、資料流計算與特殊物理加速器的統一研究框架；其價值必須由端到端基準、重組成本、精確度與能效證據共同決定。

關鍵詞：可組合核心、核心融合、資料流架構、CGRA、耦合振盪器、Kuramoto 模型、chiplet、UCIe、CXL、異質運算、可重構微架構

SECTION

1. 問題重新定義

1.1 多核心並沒有消除單執行緒瓶頸

增加核心數量可以提高吞吐量，但不會自動縮短一條依賴鏈。對指令依賴圖

$\text{cal-G}_I = (V_I, E_I),$

若一條路徑上的每個節點都依賴前一節點，增加更多執行單元並不能把該路徑任意壓縮。單執行緒性能受到至少五類因素共同限制：

- 可被揭露的指令級並行度；
- 分支與控制依賴；
- 記憶體延遲與記憶體級並行度；
- 前端取指、解碼與指令視窗容量；
- 功率、熱與時序裕度。

因此，「其他核心閒置」不等於「那些核心的全部硬體都能直接轉化為單執行緒加速」。真正問題是：

哪些資源可以共享、借用、重組或專用化？其協同成本是否低於收益？

1.2 核心融合並非全新概念

Core Fusion、Composable Lightweight Processors、MorphCore 等研究已探索過將多個簡單核心組合成較大的邏輯處理器，或讓同一微架構在高單執行緒性能與高吞吐量模式間轉換。SynCore 不應宣稱首次提出「多核融合為大核」。

新版 SynCore 的差異性只能建立在更具體的統合問題上：

- 融合模式如何與數位資料流加速器共存；
- 物理振盪器計算如何被限制在合法問題類別；
- 如何由 O-Chip、DEO、DPCPS 與 DryCore 提供跨層承諾；
- 如何以精確狀態、回退與證據制度管理重組風險；
- 如何用同一個軟體中介表示描述不同執行域。

1.3 「流動」不是對布林計算的否定

數位邏輯、資料流機器與耦合振盪器是不同抽象層：

- 布林邏輯描述離散數位電路；
- 資料流描述運算何時因輸入就緒而觸發；
- 振盪器計算利用連續時間物理動力學搜尋穩定態或近似解。

它們可以共同存在，但不能由「宇宙是流動的」推出「NOT 閘不存在」或「通用 CPU 應被振盪器取代」。

SynCore v2.1 將「流動模式」重新分成：

- 數位流動模式：可精確、可重放的資料流或空間運算；
- 物理想流動模式：帶有噪聲、校準與近似性的特殊用途運算。

SECTION

2. 設計原則

原則一：先辨識工作負載結構，再選擇硬體模式

系統不以應用名稱判定模式，而使用可量測特徵：

```
w
=
(
  p_(ILP),
  p_(TLP),
  p_(MLP),
  b_(branch),
  m_(reuse),
  d_(graph),
  q_(precision),
  r_(regularity),
  s_(state)
).
```

它們分別表示可揭露的指令級、執行緒級與記憶體級並行，分支行為、資料重用、圖結構、精度要求、規則性與狀態大小。

原則二：融合不能消除真實依賴

融合模式可以擴大指令視窗、執行單元池和記憶體請求容量，但不能令因果依賴消失。性能提升應以臨界路徑、前端、後端與記憶體限制共同估算。

原則三：物理流動只服務可合法映射的問題

只有當問題可映射到相位耦合、能量函數、儲備池動力學或其他已定義物理模型時，才允許送入 Oscillator Compute Tile。任何通用程式若沒有合法映射，必須留在純量或數位資料流域。

原則四：精確狀態與安全否決不可被類比計算取代

類比或近似加速器的輸出必須經過資格判定。記憶體保護、精確例外、ECC、熱關機、電壓護欄與安全邊界仍由數位硬體控制。

原則五：重組本身是一項成本

模式切換不是免費的一個 bit。它可能包括排空流水線、保存狀態、重設路由、載入配置、暖機、校準與驗證。

SECTION

3. SynCore v2.1 頂層架構

3.1 五類計算與控制元件

3.1.1 Scalar Tile

Scalar Tile 是相容性基線，包含：

- 取指、解碼與分支預測；
- 亂序或順序執行後端；
- 暫存器重命名；
- L1 快取與 TLB；

-
- 精確例外；
 - 向量或矩陣擴展。

它可以獨立執行，也可以加入 Fusion Cluster。每個 Scalar Tile 不必同時攜帶完整振盪器路徑。

3.1.2 Fusion Cluster

Fusion Cluster 不是把多顆現成 CPU 的 L1、暫存器與 ALU 在執行中「接線合併」，而是製造時就設計為可組合叢集。可選結構包括：

- 一個共享前端配多個後端；
- 分散式前端配統一重命名和提交；
- 分散式執行視窗與 banked physical register file；
- clustered execution；
- helper engine；
- runahead 或資料存取／執行解耦；
- banked cache 與明確資料擁有權。

邏輯上，Fusion Cluster 可形成一個較寬的執行域，但物理延遲、旁路網路、喚醒—選擇複雜度和導線功耗仍限制規模。

3.1.3 Digital Flow Array

Digital Flow Array (DFA) 是一組可重構空間單元，適合：

- 規則迴圈；
- stencil；
- 張量與矩陣核心；
- DSP；
- 串流；
- 圖子結構；

- 可靜態或半靜態排程的運算圖。

對節點 v ，觸發條件可表示為：

```

fire(v,t)
=
1
[
bigwedge_(u∈pred(v))
valid_(u→v)(t)
∧
resource_v(t)
].

```

當輸入 token 與執行資源均就緒時，節點執行：

```

y_v
=
f_v
(
{x_(u→v)},
θ_v
).

```

這是一種數位資料流語義，不需要把它描述成量子或非布林本體。

3.1.4 Oscillator Compute Tile

Oscillator Compute Tile (OCT) 是選配的特殊加速器。其基本動力學可採耦合相位模型：

```

(dθij)/(dt)
=
ωij
+
Σ_(j=1)^N
K_(ij)sin(θij-θij)
+
uij(t)

```

其中 ω_{ij} 是本徵頻率， K_{ij} 是耦合， u_{ij} 是控制或注入訊號， ξ_{ij} 表示噪聲與製程擾動。

OCT 可能服務：

- Ising／QUBO 類組合最佳化；
- 相位聚類與同步偵測；
- 某些儲備池計算；
- 可證明具有穩定吸引子的特定注意力或關係計算；
- 物理系統模擬的代理模型。

它不承諾全域最優，也不等同通用計算機。

3.1.5 Verification and Commit Engine

原稿的 Settlement Core 改為 Verification and Commit Engine (VCE)。它負責：

- 精確架構狀態；
- retirement／commit；
- 檢查點與回滾；
- 跨域資料格式與精度檢查；
- 類比結果的可接受性判定；
- 錯誤重放；
- 安全與權限；
- 證據日誌。

VCE 不是固定只佔 3% 面積且大部分時間睡眠的帳本核心；其規模取決於提交寬度、可靠度、安全需求與加速器數量。

3.2 三層互連

片上 SynFabric

片上低延遲網路用於：

- Scalar Tile 之間的融合控制；
- banked register／cache；
- DFA token 傳遞；
- VCE 提交與回退；
- OCT 設定與讀出。

片上路徑必須使用實際 NoC、crossbar、mesh、ring、hierarchical bus 或專用短距離鏈路評估，不能以「拓撲連接不等於物理距離」取消導線延遲。

封裝內晶粒互連

封裝內可使用 UCle 類 die-to-die 介面或自訂短距離鏈路。UCle 3.0 已支援更高資料率與管理功能，適合作為異質晶粒整合的介面基礎；但其通訊仍有 PHY、協議、緩衝與錯誤處理成本，不能視為單週期共享暫存器。

系統級 CXL

CXL 適合：

- 記憶體擴展與池化；
- 一致性設備；
- 加速器與主機資料交換；
- 系統級資源組合。

截至 2026 年，CXL 4.0 已提高資料率並增強連接與 RAS，但它仍是系統級互連。SynCore 不再以 CXL 實現跨機箱或任意距離的單一超寬 CPU 後端。

SECTION

4. 四種執行模式

4.1 I-Mode：獨立核心模式

各 Scalar Tile 獨立執行不同執行緒，DFA 與 OCT 依需求休眠或服務其他任務。此模式提供最穩定的通用吞吐量與軟體相容性。

4.2 F-Mode：彈性融合模式

數個 Scalar Tile 形成 Fusion Cluster，共同服務一條或少數高優先級執行緒。

可組合的資源包括：

- 指令視窗容量；
- 執行單元；
- load/store queue；
- MSHR；
- cache bank；
- 預取器；
- 分支或 runahead 輔助資源；
- 功率與熱預算。

不一定可組合的資源包括：

- 距離過遠的單週期旁路；
- 任意核心私有 L1 的瞬間統一；
- 跨封裝的精細喚醒—選擇；
- 不受依賴約束的 IPC。

4.3 D-Mode：數位流動模式

編譯器將可提取區域映射到 DFA。通用核心處理控制、不規則部分與例外，DFA 處理規則資料流。

4.4 P-Mode：物理流動模式

可映射的最佳化、同步或類比計算被送入 OCT。VCE 必須檢查：

- 映射是否有效；
- 讀出是否收斂；
- 解的品質；
- 重複執行一致性；
- 是否需要數位精修；
- 是否超出校準域。

4.5 H-Mode：混合管線模式

真實工作負載可同時使用多個執行域。例如：

控制與序列邏輯

→

DFA 張量／串流核心

→

OCT 近似最佳化

→

Scalar／VCE 驗證。

混合模式的性能必須用端到端延遲與資料搬移成本判定，不能把各子模組的最佳倍率直接相乘。

SECTION

5. 融合模式的性能邊界

5.1 指令依賴圖與臨界路徑

令單執行緒包含 N 個動態指令，有效發射寬度為 W_{eff} ，依賴圖臨界路徑長度為 $C(\text{cal-G}_I)$ ，記憶體瓶頸下界為 T_{mem} ，則融合模式執行時間至少滿足：

$$\begin{aligned}
& T_F \\
& \geq \\
& \max \\
& (\\
& C(\text{cal-G}_I), \\
& (N)/(W_{\text{eff}}), \\
& T_{\text{mem}}, \\
& T_{\text{front}} \\
&) \\
& + \\
& T_{\text{coord}} \\
& + \\
& T_{\text{reconfig}}.
\end{aligned}$$

因此：

$$\begin{aligned}
& S_F \\
& = \\
& (T_{\text{base}})/(T_F)
\end{aligned}$$

受到依賴、前端、記憶體、協同和重組共同限制。

5.2 為什麼 4 核融合不等於 4 倍 IPC

喚醒—選擇邏輯、暫存器讀寫埠、旁路網路與線路延遲通常隨寬度非線性增加。可用執行單元多於就緒指令時，增加資源沒有收益。

只有在以下條件同時成立時，融合才可能顯著受益：

- 工作負載具有足夠 ILP 或 MLP；
- 指令視窗能揭露這些並行；
- 前端與分支不成為主要瓶頸；
- 記憶體系統能支撐額外請求；

- 重組與協同成本可被工作區間攤銷；
- 功率與熱允許額外單元活化。

5.3 多路徑推測的限制

同時執行多個分支路徑會消耗：

- 取指與解碼頻寬；
- rename／ROB 容量；
- cache 和 TLB；
- 記憶體頻寬；
- 功率。

因此它只適合高代價、低分支扇出的特定區域。新版不再宣稱遇到 if-else 就同時執行所有可能路徑。

5.4 動態指令融合的合法位置

把

$(a+b)(c-d)$

變成一個內部複合操作，可能透過：

- 編譯器 pattern；
- macro-op fusion；
- micro-op fusion；
- JIT；
- dataflow region；
- 自訂指令；

實現。但硬體仍要處理依賴、例外、溢位、精度與退休語義。SynCore 的貢獻應是提供可映射區域與資源，不是宣稱任意指令序列可免費融合。

6. 流動模式的兩條技術路線

6.1 數位資料流：可重現與可程式化

TRIPS、EDGE、CGRA 與 Plasticine 等工作顯示，將運算圖空間映射到硬體，可降低集中式暫存器、動態調度與資料搬移成本。SynCore 的 DFA 可以借鑑這些路線，但必須處理：

- 不規則控制；
- 動態記憶體別名；
- 配置容量；
- 路由壅塞；
- 中間資料儲存；
- 例外；
- 編譯器映射時間；
- 與通用核心一致性。

6.2 耦合振盪器：特殊問題的物理求解器

耦合振盪器硬體已有 Ising 機、最佳化晶片與多種 CMOS／奈米振盪器原型。它們證明「相位與耦合可以承載計算」，但同時也揭示：

- 問題嵌入成本；
- 耦合矩陣可程式化；
- 製程變異；
- 噪聲；
- 局部最小值；

- 讀出與校準；
- 數位基線比較；

是不可省略的部分。

對目標函數 $E(s)$ ，只有存在明確映射

cal-M:

$E(s)$

| ->

(

ω ,

K,

u,

cal-R

)

時，OCT 才是合法後端。cal-R 是讀出規則。

6.3 Phase-LM 的地位

Phase-LM 或「以同步取代部分注意力」可保留為應用研究方向，但不能作為 SynCore 成立的前提。近期有研究探索 Kuramoto 類動力學與注意力、圖神經網路或物理基底的關係，但其適用資料集、模型品質、硬體映射和端到端能耗仍需要獨立驗證。

SynCore 的商業價值不能建立在「 10^8 個相位單元等效 10^{11} 個參數」這種未定義等價上。

SECTION

7. 模式切換與狀態管理

7.1 切換不是單一 bit

完整模式切換採用狀態機：

Observe

→

Decide

→

Quiesce

→

Checkpoint

→

Drain

→

Reconfigure

→

Warm

→

Validate

→

Run.

切換時間：

$T_{\text{(switch)}}$

=

$T_{\text{(quiesce)}}$

+

$T_{\text{(checkpoint)}}$

+

$T_{\text{(drain)}}$

+

$T_{\text{(route)}}$

+

$T_{\text{(load)}}$

+

$T_{\text{(warm)}}$

+

$T_{\text{(validate)}}$.

只有 clock gating、局部單元喚醒等低階動作可能非常快速；完整語義模式切換可能落在微秒至毫秒尺度，視狀態量與配置而定。

7.2 Context Vault

原 Q-Storage 改為 Context Vault，儲存：

- 架構暫存器；
- rename map；
- PC 與控制狀態；
- 必要的執行視窗摘要；
- TLB／cache hint；
- 加速器配置；
- 檢查點與版本；
- 可信度與錯誤資訊。

它不承諾在外部記憶體中保存完整 L1 的所有時序語義，也不以「量子態」命名。

7.3 精確狀態

VCE 保證：

- 例外看見一致的架構狀態；
- 錯誤的推測不外洩；
- 類比結果不直接覆寫關鍵狀態；
- 模式失敗可回到安全檢查點；
- 加速器間的數值格式與捨入可追蹤。

SECTION

8. SynIR：統一中介表示

8.1 區域標註

SynIR 將程式切分為區域：

@scalar

@fusion_candidate

@dataflow

@oscillator_approx

@verify

每個區域攜帶：

- 輸入與輸出；
- 依賴圖；
- 精度；
- 容錯；
- 時限；
- 可重放性；
- 狀態大小；
- 預估搬移量；
- 合法後端。

8.2 執行合約

對區域 r ，定義合約：

```
cal-C_r  
=  
(  
   $\epsilon_r$ ,  
   $\tau_r$ ,  
   $P_r$ ,  
   $T_r$ ,  
   $\rho_r$ ,
```

其中：

- ϵ_r ：誤差容忍；
- τ_r ：期限；
- P_r ：功率上限；
- T_r ：熱上限；
- ρ_r ：可靠度；
- χ_r ：可回退性。

O-Chip 根據合約與即時資源選擇執行域。

8.3 舊二進位相容性

既有二進位預設在 Scalar Tile 執行。只有經過：

- profile；
- JIT；
- 動態二進位翻譯；
- 明確 API；
- 編譯器分析；

確認的區域，才可進入融合或流動模式。新版不保證未修改的老遊戲自動獲得 4-6 倍幀率。

SECTION

9. 多目標編排模型

工作負載圖為：

$$\text{cal-G}_W$$
$$=$$
$$(V_W, E_W, a_W),$$

資源圖為：

$$\begin{aligned} & \text{cal-G_R} \\ & = \\ & (V_R, E_R, a_R). \end{aligned}$$

模式映射 Π 同時指定：

- 區域到執行域；
- 資料位置；
- Fusion Cluster 規模；
- DFA 配置；
- OCT 耦合；
- DEO 包絡；
- DPCPS 功率承諾；
- DryCore 熱承諾；
- 回退路徑。

目標函數：

$$\begin{aligned} & \min_{\Pi} \\ & J(\Pi) \\ & = \\ & \alpha L_{\text{(end)}} \\ & + \\ & \beta E_{\text{(sys)}} \\ & + \\ & \gamma A_{\text{(active)}} \\ & + \\ & \delta D_{\text{(move)}} \\ & + \\ & \eta C_{\text{(reconfig)}} \end{aligned}$$

約束包括：

$$P(t) \leq P_{\text{cert}}(t),$$

$$T_{\text{max}}(t) \leq T_{\text{i,max}},$$

$$B_{\text{max}}(t) \leq B_{\text{max}},$$

$$\text{err}(r) \leq \varepsilon_r,$$

state_(commit)

∈

cal-S_(precise).

SECTION

10. 與 EveMissLab 計算架構系列的關係

10.1 O-Chip

O-Chip 觀察任務、系統遙測與應用意圖，建立跨域計畫。SynCore 不自己承擔整個系統級意圖推斷。

10.2 DEO

DEO 決定當前可以合法啟用的運作包絡。Fusion Cluster 的寬度、DFA 活化範圍與 OCT 迭代時間都必須服從 DEO。

10.3 DPCPS

DPCPS 提供局部功率、電流瞬變與 PDN 證書。SynCore 不能因「融合功率預算」而假設可以突破供電限制。

10.4 DryCore

DryCore 提供熱阻、冷卻容量、露點與失效狀態。熱輪換不能創造額外冷卻能力，只能重新分配時空熱負載。

10.5 SDMCA

SDMCA 提供封裝外模組、機箱與互連拓撲。片內融合仍需專門的物理設計；樓梯或環形機箱不能直接縮短核心旁路路徑。

10.6 統一資料流

應用／SynIR

→

O-Chip 編排

→

DEO／DPCPS／DryCore 證書

→

SynCore 執行域

→

VCE 驗證與提交。

SECTION

11. 失效模式與安全

失效模式 可能後果 偵測 回退

Fusion 協同延遲過高 IPC 下降、功耗增加 硬體計數器 拆回獨立核心

DFA 路由壅塞 延遲與能耗惡化 token queue 重映射或 CPU

OCT 不收斂 錯誤或不穩定解 order parameter／殘差 數位求解器

OCT 校準漂移 解品質降低 calibration test 重校準／停用

Context Vault 損壞 無法恢復 ECC／checksum 最近安全檢查點

VCE 壅塞 提交停頓 commit queue 降低加速器併發

UCle link 降級 晶粒頻寬下降 link health 降速、繞路、隔離

CXL 不可用 遠端記憶體失聯 fabric manager 本地副本或中止

功率證書失效 droop／降頻 DPCPS 遙測 縮小包絡

熱證書失效 過熱 DryCore 遙測 降載或關機

類比與近似模式不得用於未經資格化的安全關鍵控制路徑。

SECTION

12. 驗證方法

12.1 證據等級

- E0：架構與可證偽假說；
- E1：分析模型與軟體模擬；
- E2：FPGA／電路板子系統原型；
- E3：多子系統硬體在迴路；
- E4：專用晶片或 chiplet 原型；
- E5：跨批次、跨實驗室與真實工作負載重現。

本篇為 E0。

12.2 V0：軟體與微架構模擬

工具候選：

- gem5；
- Sniper；
- LLVM／MLIR；
- CGRA mapper；

-
- SPICE／Verilog-A ；

- 熱與 PDN 模型 。

比較：

- 一般 out-of-order 核心 ；
- 多個獨立核心 ；
- Core Fusion 類基線 ；
- SynCore Fusion Cluster ；
- GPU／CGRA ；
- SynCore DFA ；
- 數位 Ising／最佳化基線 ；
- OCT 模型 。

12.3 V1：FPGA 數位流動原型

實作：

- RISC-V 控制核 ；
- 小型 DFA ；
- SynIR runtime ；
- VCE ；
- Context Vault ；
- 部分重配置 。

量測：

- kernel latency ；
- energy estimate ；

- route utilization ;
- configuration time ;
- end-to-end speedup ;
- 回退正確性。

12.4 V2：可組合 RISC-V 叢集

設計兩至四個簡單 Scalar Tile，逐步加入：

- 共享 instruction window ;
- banked register file ;
- 分散式 load/store queue ;
- helper/runahead ;
- 統一 commit。

目標不是先追求極寬 IPC，而是測出：

收益

-

協同成本

隨融合規模的曲線。

12.5 V3：振盪器板級原型

建立可程式耦合振盪器陣列，選擇小型 QUBO／Max-Cut／聚類問題，比較：

- time-to-solution ;
- solution quality ;
- energy-to-solution ;
- 重複性；

-
- 校準時間；
 - 問題嵌入成本；
 - 數位精修成本。

12.6 V4：chiplet 原型

只在 V0–V3 顯示明確收益後，才進入：

- Scalar／Fusion die；
- DFA die；
- OCT die；
- VCE／I/O die；
- UCIe 封裝；
- CXL 系統連接。

SECTION

13. 基準工作負載

融合模式

- SPEC CPU 類單執行緒工作；
- 編譯器前端；
- 模擬器核心；
- pointer chasing；
- branch-heavy kernel；
- memory-level-parallel microbenchmark；

-
- JIT／腳本執行。

數位流動模式

- GEMM ；
- stencil ；
- FFT ；
- sparse linear algebra ；
- graph traversal ；
- DSP ；
- packet／stream processing 。

物理流動模式

- Max-Cut ；
- QUBO ；
- graph coloring ；
- 相位聚類 ；
- 小型同步注意力研究模型 ；
- reservoir computing 。

混合模式

- 遊戲或模擬器的控制＋AI／物理子系統 ；
- 圖形前處理＋張量推論＋最佳化 ；
- compiler／database pipeline ；
- agentic workload 中的控制、檢索、推論與排程。

所有比較必須包含資料搬移、配置、校準、讀出與驗證，不得只比較核心運算時間。

SECTION

14. 可證偽命題

H1：融合收益存在但非線性

對具有足夠 ILP／MLP 且工作區間可攤銷重組成本的負載，Fusion Cluster 的端到端性能優於等面積獨立小核；但增益隨融合規模呈飽和或下降。

H2：數位流動適合規則圖

對可靜態或半靜態映射的規則運算，DFA 的能量—延遲積優於通用 out-of-order 核心；對不規則控制或高動態別名工作負載，優勢下降或消失。

H3：振盪器只在特定映射下有優勢

對可直接映射到耦合振盪能量景觀的問題，OCT 在相同解品質下可能降低 time-to-solution 或 energy-to-solution；若包含嵌入、校準與數位精修後仍無優勢，則 P-Mode 不成立。

H4：混合模式不保證乘法增益

H-Mode 只有在跨域搬移、模式切換與驗證成本低於專用域收益時，才優於最佳單一後端。

H5：跨晶粒融合存在明確邊界

封裝內互連可以支援粗粒度或階層式融合，但當協同延遲接近核心喚醒—選擇或旁路時間尺度時，精細單執行緒融合失去收益。CXL 更適合記憶體與設備組合，而非細粒度超純量後端。

SECTION

15. 近期 MVP

MVP 1：SynIR 與成本模型

先完成：

- SynIR 區域標註；
- workload feature extractor；
- backend cost model；
- mode selector；
- trace replay；
- 可視化報告。

輸出每個區域的：

- 合法後端；
- 預估延遲；
- 資料搬移；
- 重組成本；
- 能耗；
- 不確定性；
- 回退策略。

MVP 2：Fusion 模擬器

在 gem5 或自建 trace simulator 中實作：

- 2/4 tile cluster；
- banked execution units；
- distributed instruction window；
- variable fusion size；
- Context Vault；

- VCE。

MVP 3：FPGA DFA

將一組 MLIR/LLVM kernel 映射到 FPGA CGRA，與 CPU/GPU 比較端到端性能。

MVP 4：OCT 電路板

以可程式耦合振盪器板驗證小型最佳化問題，建立公開校準與重現資料。

這四個 MVP 可以平行推進，不需要先製造完整 SynCore 晶片。

SECTION

16. 技術地位與公開聲明

SynCore v2.1 不宣稱：

- 首次發明核心融合；
- 多核融合必然線性提升 IPC；
- 未修改舊遊戲必然提升數倍幀率；
- CXL 可建立無限、單週期或跨機櫃超級核心；
- Hybrid Core 只用一個 bit 就能完整切換語義；
- Flow Path 固定只耗 Boolean Path 的十分之一；
- Kuramoto 同步可替代任意 CPU/GPU 工作；
- 類比運算自然比數位精確；
- Phase-LM 已證明可以取代 Transformer；
- 流動本體論已被硬體證明。

SynCore v2.1 的公開主張是：

把可組合單執行緒資源、數位資料流與特殊物理加速器納入同一個可回退、可量測、可證偽的異質架構，是一條值得工程驗證的研究路線。

SECTION

17. 結論

SynCore 的早期版本包含兩個有價值但彼此不同的直覺：

- 閒置或分散的硬體資源，能否在需要時組合成更適合單執行緒的執行域；
- 某些問題能否不經完整的通用指令序列，而直接映射到資料流或物理動力學。

v2.1 不再用「同一個吸引子」證明兩者必然是同一種核心，而是以工程分層使它們能夠共存：

通用純量

+

彈性融合

+

數位資料流

+

物理流動

+

精確驗證。

這個架構的成敗不取決於宣言是否宏大，而取決於五個問題：

- 融合後真正揭露了多少 ILP／MLP；
- 資料流映射是否降低了搬移與調度成本；
- 振盪器的解品質是否值得校準與讀出代價；
- 混合管線是否優於最佳單一後端；
- 系統能否在失效時安全、快速、可追蹤地回退。

SynCore 因此從「單核至尊」與「流動本體論晶片」轉為一個更嚴格的研究計畫：同步不是神秘名稱，融合也不是倍率承諾；它們是必須被時序、功率、互連、編譯器與證據共同約束的系統能力。

參考文獻

- E. Çipek, M. Kırman, N. Kırman, J. F. Martínez, “Core Fusion: Accommodating Software Diversity in Chip Multiprocessors,” ISCA, 2007.
<https://people.ece.cornell.edu/martinez/doc/isca07.pdf>
- C. Kim et al., “Composable Lightweight Processors,” MICRO, 2007.
<https://www.cs.utexas.edu/ftp/dburger/papers/MICRO07.pdf>
- Khubaib et al., “MorphCore: An Energy-Efficient Microarchitecture for High Performance ILP and High Throughput TLP,” MICRO, 2012.
https://hps.ece.utexas.edu/people/miladh/pub/morphcore_micro2012.pdf
- The TRIPS Project, “Explicit Data Graph Execution Architecture and Prototype.”
<https://www.cs.utexas.edu/~cart/trips/overview.html>
- R. Prabhakar et al., “Plasticine: A Reconfigurable Architecture for Parallel Patterns,” ISCA, 2017.
<https://csl.stanford.edu/~christos/publications/2017.plasticine.isca.pdf>
- G. Csaba, W. Porod, “Coupled Oscillators for Computing: A Review and Perspective,” Applied Physics Reviews, 2020.
<https://pubs.aip.org/aip/apr/article/7/1/011302/997386/>
- T. Wang, J. Roychowdhury, “Oscillator-based Ising Machine,” 2017.
<https://arxiv.org/abs/1709.08102>
- “An Ising Solver Chip Based on Coupled Ring Oscillators with a 48-node All-to-all Connected Array Architecture,” Nature Electronics, 2023.
<https://www.nature.com/articles/s41928-023-01021-y>
- “A Coupled-Oscillator-Based Ising Chip for Combinatorial Optimization,” Nature Electronics, 2025.

<https://www.nature.com/articles/s41928-025-01393-3>

- UCle Consortium, “UCle 3.0 Specification,” 2025.
<https://www.uciexpress.org/specifications>
- CXL Consortium, “CXL 4.0 Specification,” 2025.
<https://computeexpresslink.org/>
- J. A. Acebrón et al., “The Kuramoto Model: A Simple Paradigm for Synchronization Phenomena,” *Reviews of Modern Physics*, 2005.
https://scala.uc3m.es/publications_MANS/PDF/finalKura.pdf
- F. Pasqualetti, T. Guo, “Attention by Synchronization in Coupled Oscillator Networks,” preprint, 2026.
<https://arxiv.org/abs/2606.12059>

文件結束