Ενσωματωμένα Υπολογιστικά Συστήματα (Embedded Computer Systems)

Μάθημα 9

Δημήτρης Λιούπης

Περιγραφή Μαθήματος

Το μάθημα εξετάζει την,

- συ-σχεδίαση υλικού/λογισμικού (h/w – s/w codesign)
  - systems on chip
  - internet enabled controllers

Domain of Application

- Mixed hardware/software systems
- System on Chip (SoC)
- Embedded Systems
  - airplanes, cars ...
  - telecommunications
  - kitchen appliances
  - …

Goals

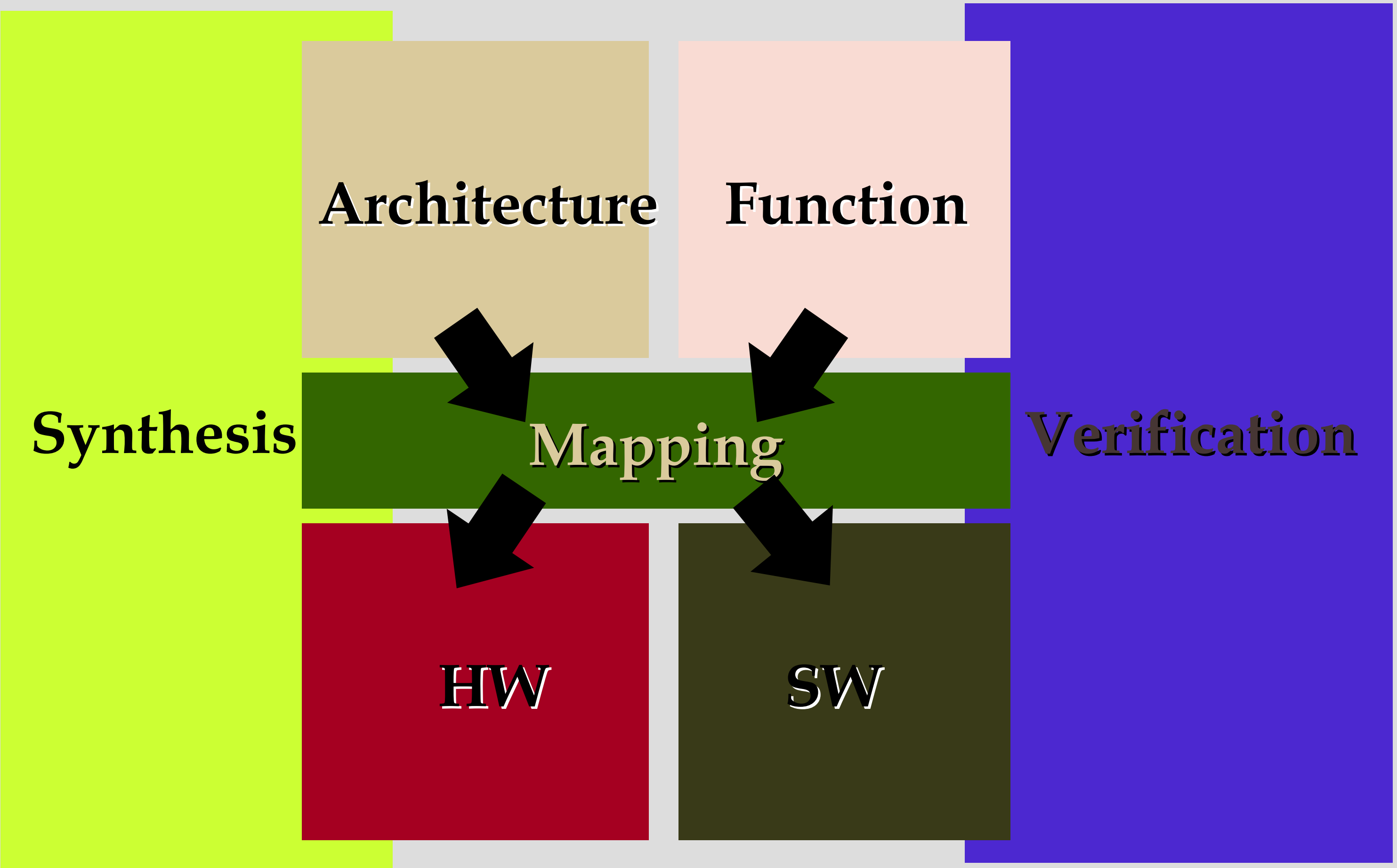
- Design/Implementation Verification

  (Formal Verification, simulation, rapid prototyping)
- Hardware, Software, and Interface Synthesis
  - different hardware and software implementation styles
- *Designer can concentrate on high level issues*

Methodology

- Separation between function, and communication
- Unified refinable formal specification model
  - –facilitates system specification
  - –implementation independent
    - ➔eases HW/SW trade-off evaluation and partitioning

Co-Design Methodology



Hardware/Software Co-Design Goals and Requirements

- Separation between function, timing and communication
- Unified formal specification model
  - facilitates system specification
  - implementation independent
    - ➔eases HW-SW trade-off evaluation and partitioning
- Design/Implementation Verification
  (Formal Verification, simulation, rapid prototyping)
- Automatic Hardware, Software and Interface Synthesis
  - different hardware and software implementation styles
  - designer can concentrate on system level issues

Codesign Finite State Machines

- We have chosen an FSM model for
  - uncommitted
  - synthesizable
  - verifiable

  control-dominated HW/SW specification
- Translators from
  - state diagrams,
  - Esterel, ECL, ReactiveJava
  - HDLs

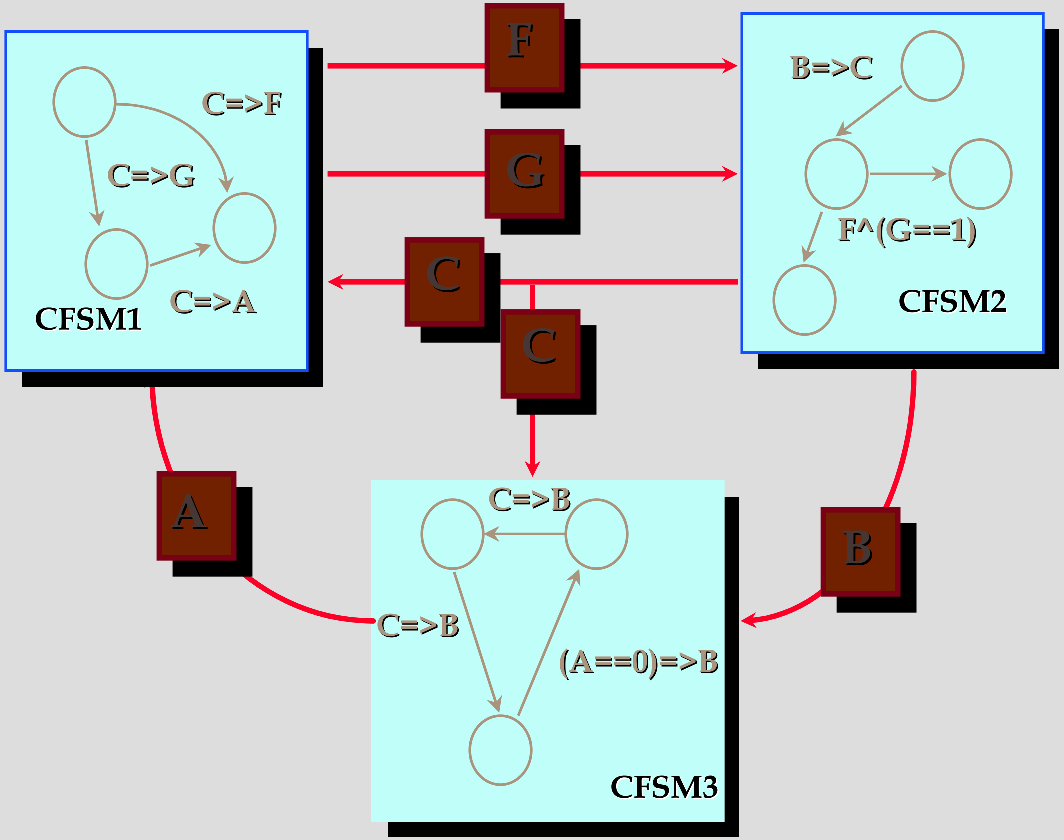
  into a single FSM-based language

CFSM behavior

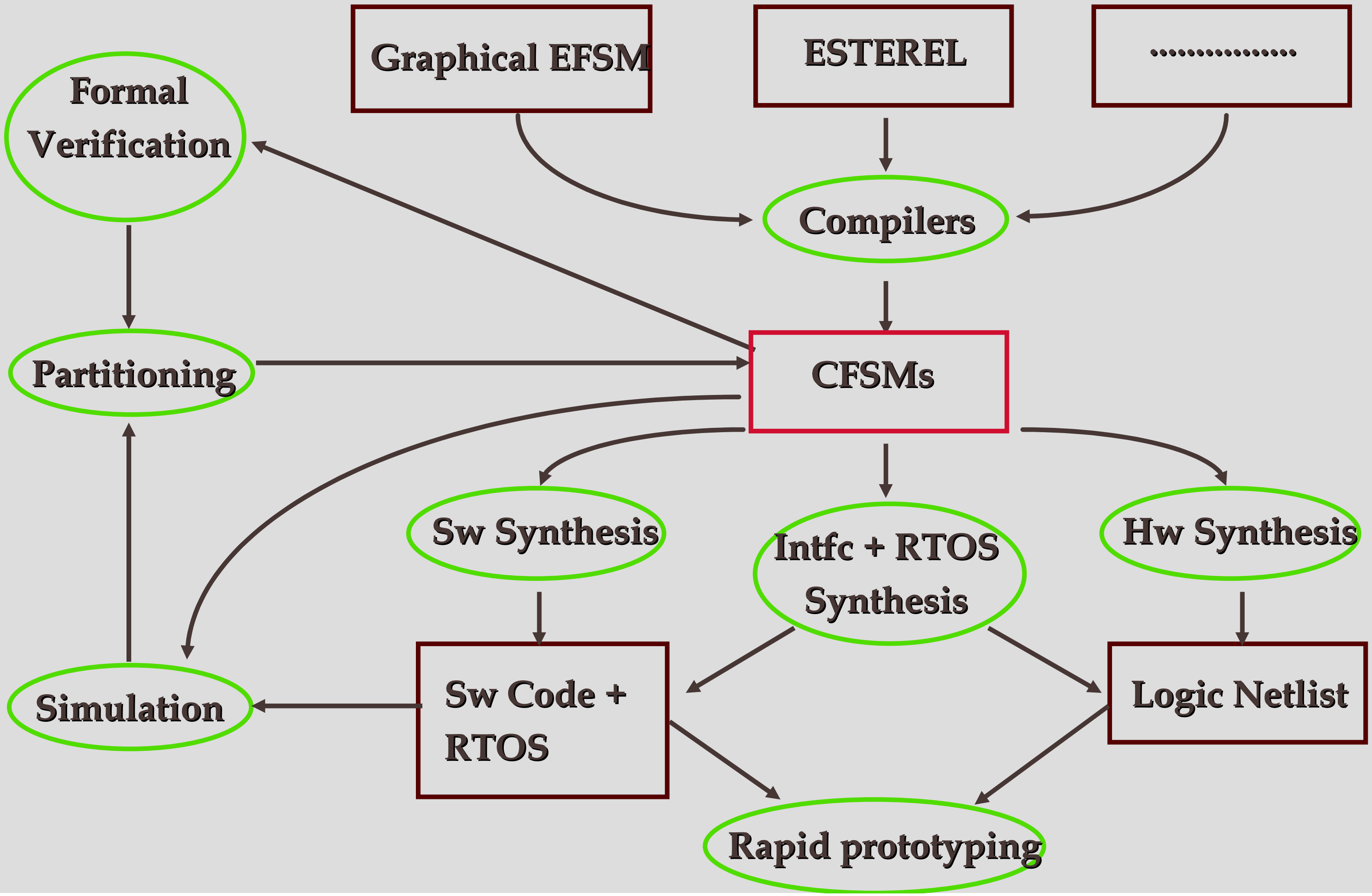
- Four-phase cycle:
  - ☆ Idle
  - 🕐 Detect input events
  - 🕑 Execute one transition
  - 🕒 Emit output events
- Software response could take a long time:
  - Unbounded delay assumption
- Need efficient hw/sw communication primitive:
  - Event-based point-to-point communication

Network of CFSMs: Depth-1 Buffers

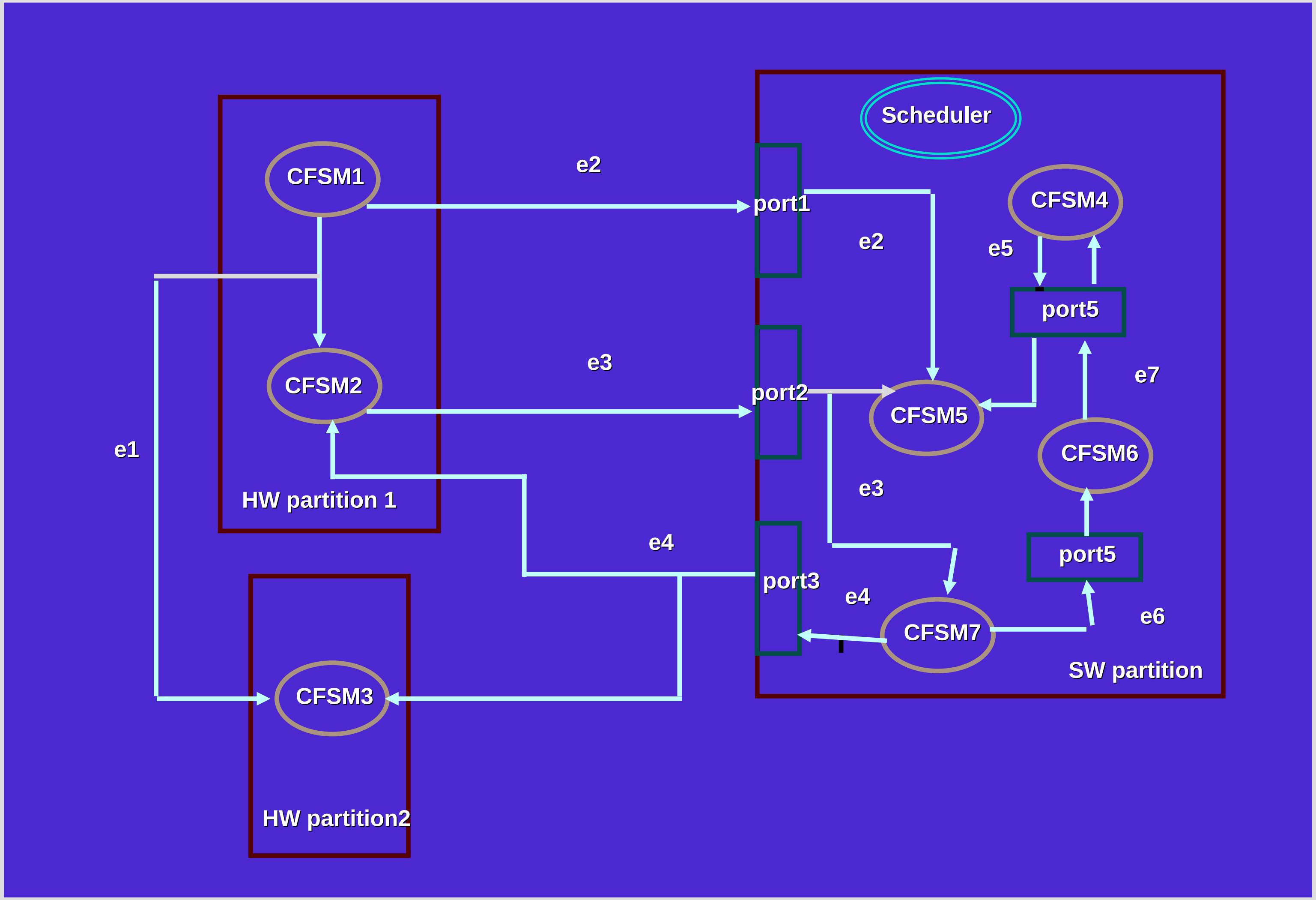
- Globally Asynchronous, Locally Synchronous (GALS) model



POLIS Codesign Methodology



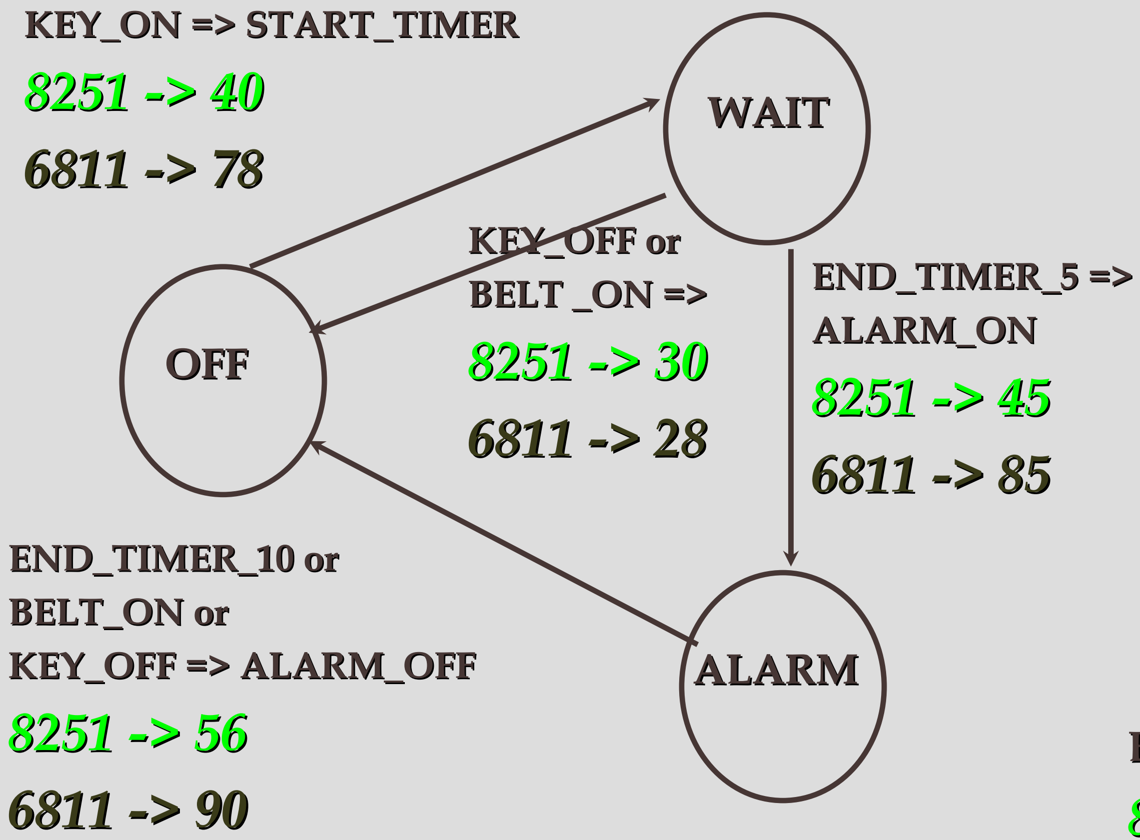
System Architecture



High-level Co-simulation

- Functional (untimed) simulation allows one to:
  - check functional (partial) correctness, by generating inputs and observing outputs
  - debug the design, by easy access to internal states
- High-level (timed) co-simulation allows one to check:
  - feasibility analysis for specification
  - hardware/software partitioning
  - architecture selection (CPU, scheduler, ...)
- Cannot be used to validate the final implementation
  - need a much more detailed model of HW and SW architecture

Simulation model annotation

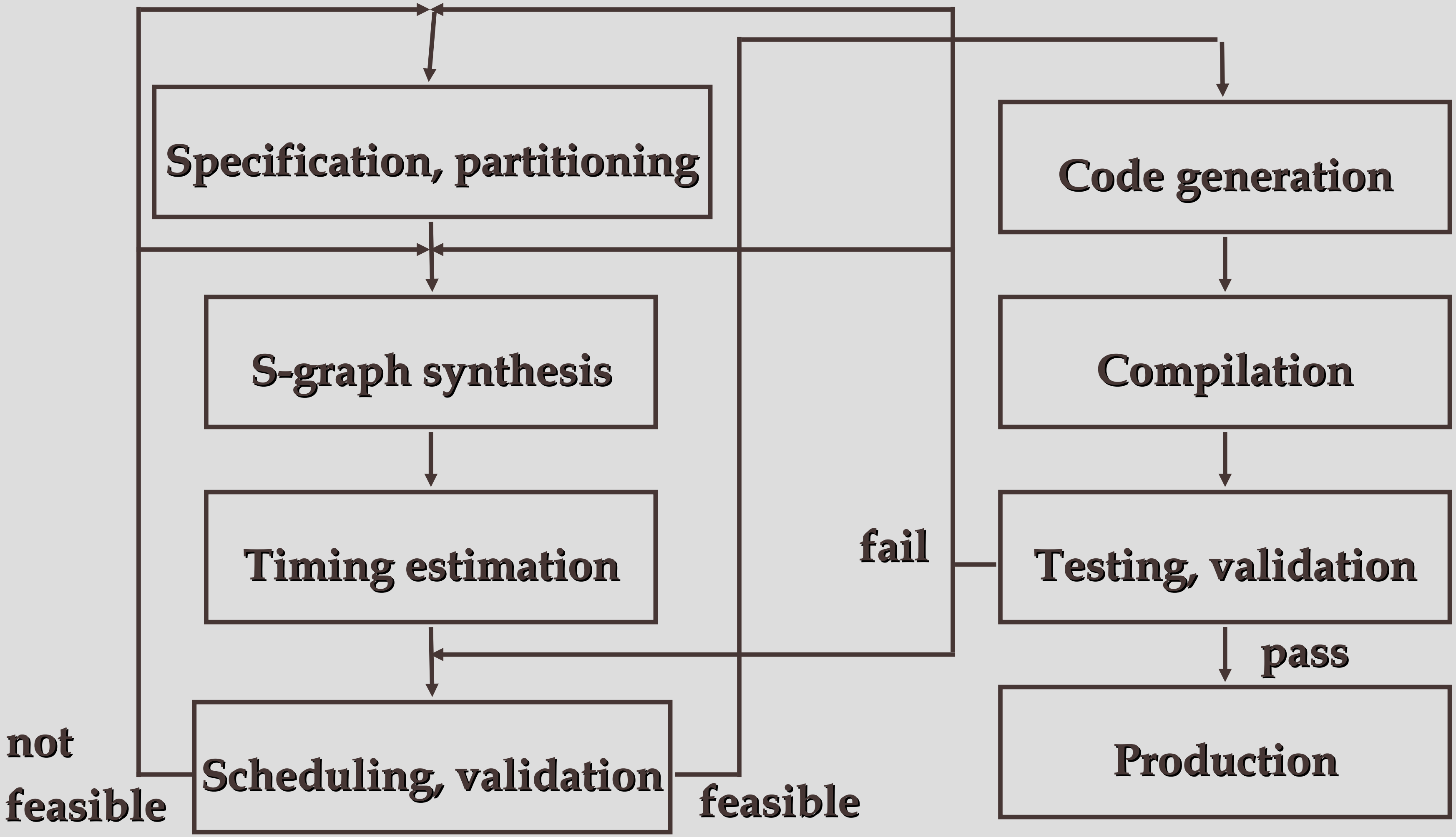


Empty execution

8251 -> 62

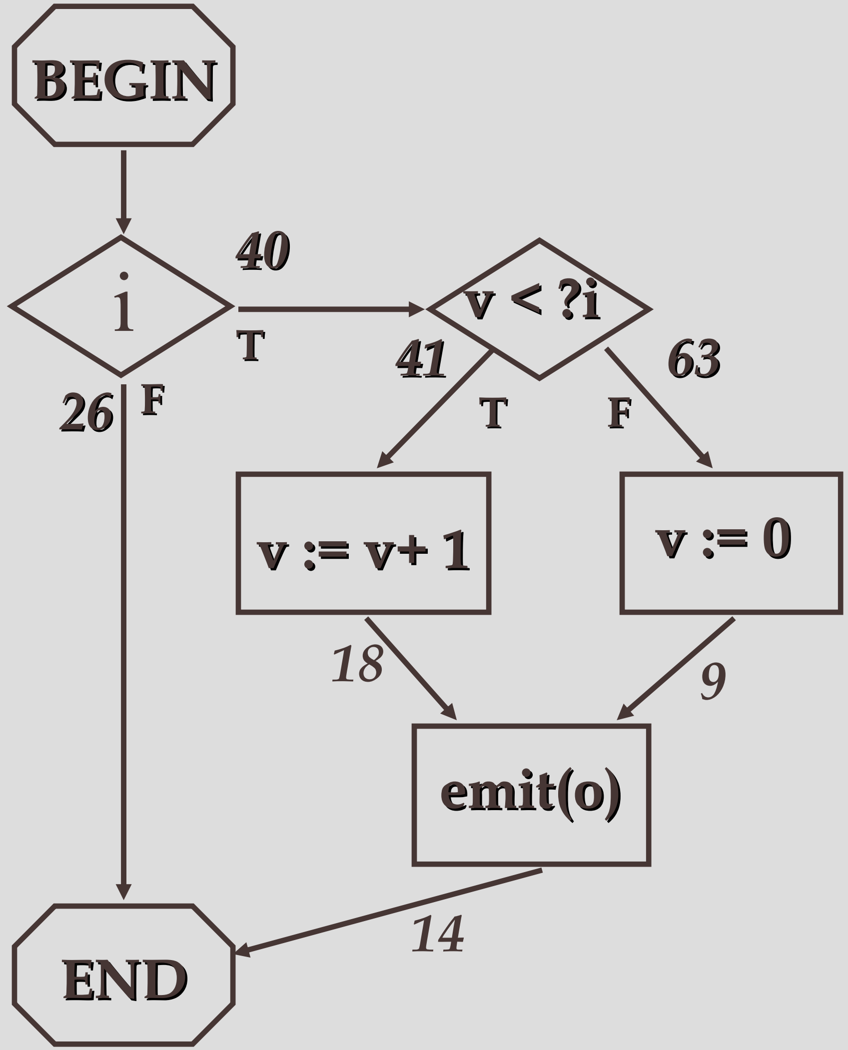
6811 -> 78

Software synthesis procedure



Software synthesis and estimation

- Example: 68HC11 timing estimation
- Cost assigned to s-graph edges
  - (different costs for taken/not taken branches)
- Estimated time:
  - min: 26 cycles
  - max: 126 cycles
- Accuracy: within 20% of profiling



Current Design Flow

- System specification:
  - Esterel, ECL
  - graphical CFSM net editor
- SW synthesis and estimation
- High-level co-simulation
  - functional debugging
  - SW+HW architecture definition and mapping
- Formal verification
- SW, HW, RTOS synthesis
- Low-level co-simulation and prototyping

Conclusions

- Embedded system design requires a new methodology
- Separation between function, communication and timing
- Functional CFSM model for hardware and software
  initially unbounded delays refined after architecture mapping
- High-level co-simulation used for architecture selection
- Partitioned hardware and software implementations automatically generated
- What next?
  - MetroPolis project
  - Cadence Virtual Component Codesign

Felix: Virtual Component