

# ВГРАДЕНИ СОФТУЕРНИ СИСТЕМИ

---



СУ „Св. Климент Охридски“  
Факултет по Математика и Информатика  
Увод в Софтуерното Инженерство

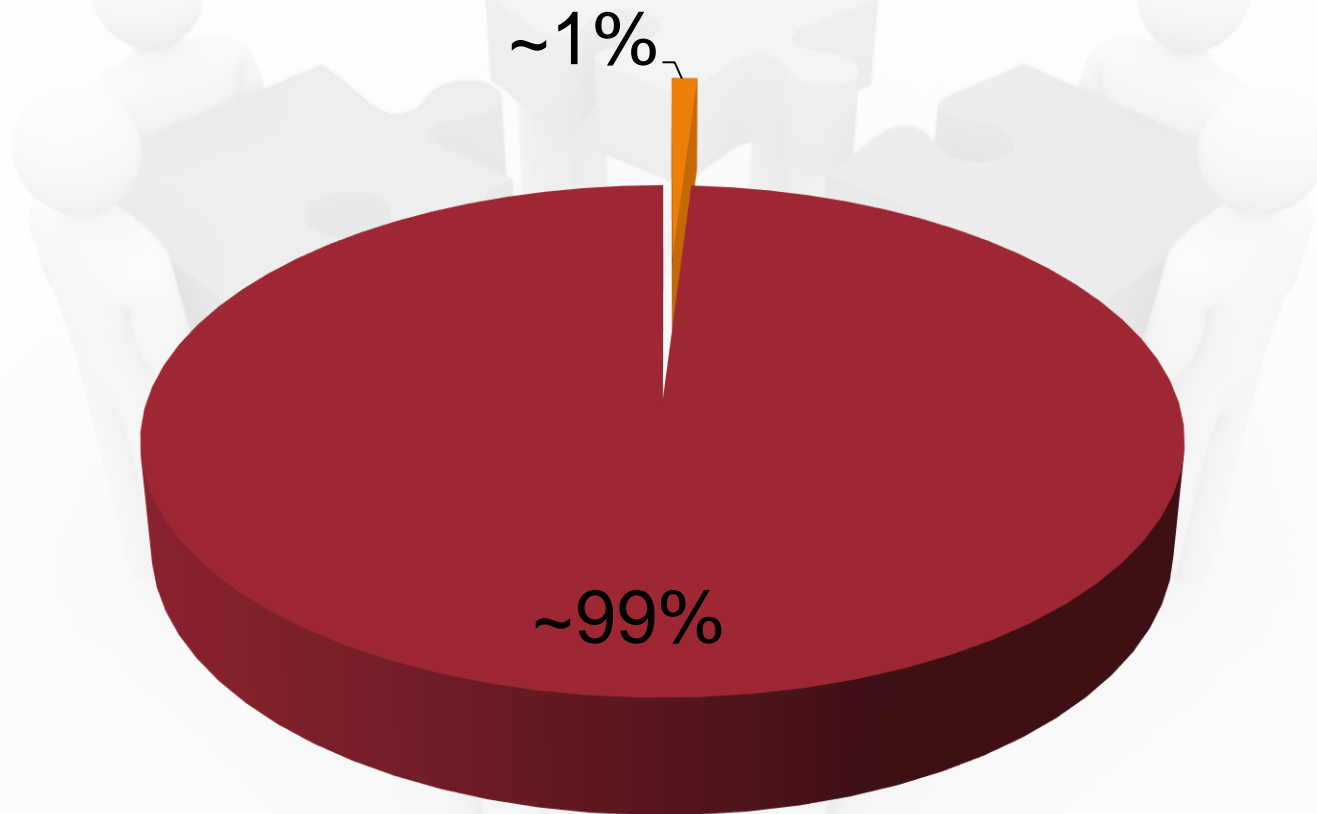
# Съдържание

- **Определение за вградени системи**
  - Основни понятия
  - Характеристики
- **Подход и процес при проектиране на ВСС**
- **Стандарт IEC 61113**
  - Програмируем логически контролер
  - Основни езици
    - Повече за Ladder diagrams
- **Системи за реално време**
  - Операционни Системи за Реално Време (ОСРВ)
  - Задачи на ОСРВ
  - Архитектура на ОСРВ

# Вградени софтверни системи



# Разпространение на вградените системи

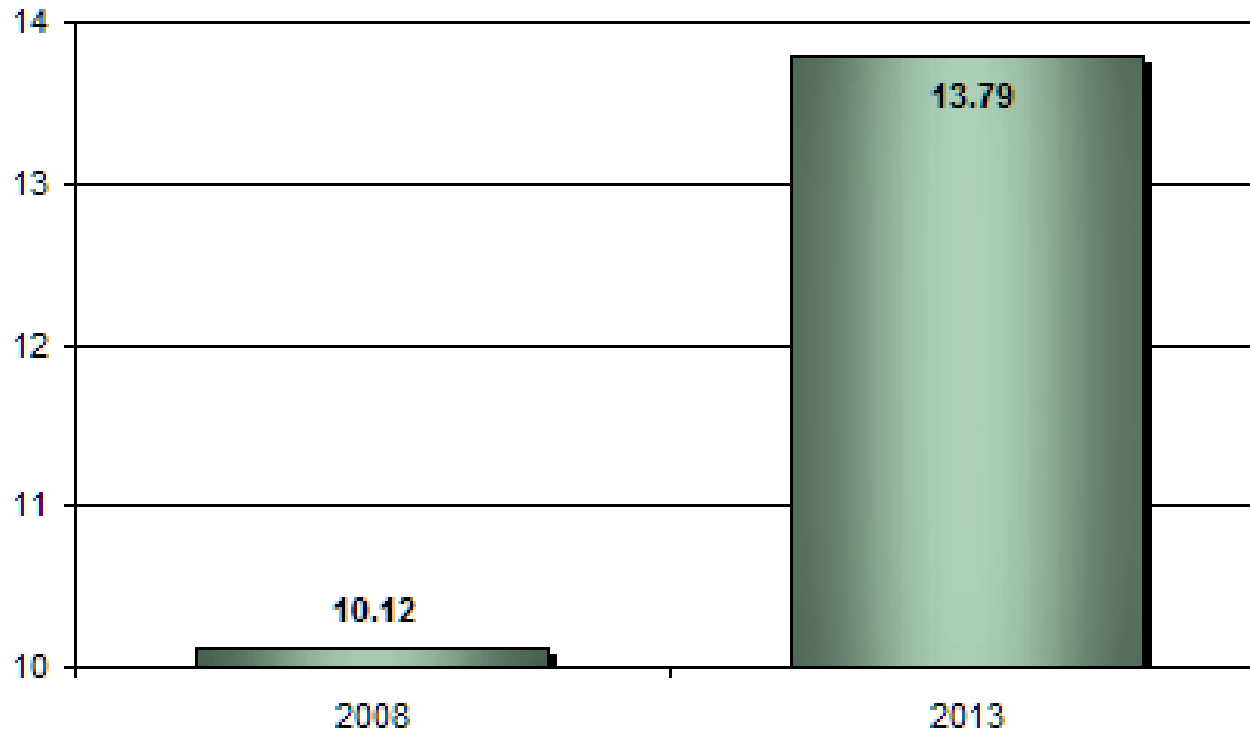


■ Работни станции    ■ Вградени процесори

# Разпространение на вградените системи

Total Market, Processors in Embedded Applications,  
Modeled Unit Shipments, 2008-2013  
(Units in Billions)

2008-2013 CAGR: 6.38%



# Вградени софтуерни системи (ВСС)

- Включват голям набор, различни по размер компютърни системи
  - Малки – софтуер за управление на домакински уред (пералня, кафе-машина и т.н.)
  - Големи – Бордови компютър в превозно средство
  - Ултра големи (включително и на няколко нива) – Управление на технологични процеси в цял завод

# Дефиниция на ВСС

- Неформално: почти всяка компютърна система, различна от настолен компютър, лаптоп или сървър
- *Смесена хардуерна/софтуерна система, която е предназначена за строго специфично приложение и е част от по-голяма физическа система, с която е свързана (поне логически) и реагира на нейните събития.*

# Вградени системи

- Критични системи/Системи за реално време
- Системи с общо предназначение





# Някои характеристики на някои ВСС

- Ниски хардуерни ресурси (памет, честота на процесора и др.)
- Работа в реално време
- Критичност по отношение на безопасността (*Safety-critical*)
  - Висока надежност
- Дълъг период на употреба на системата
- Трудност при извършване на промени
- Хардуерната архитектура на платформата, където се разработва системата може да е различна от архитектурата на платформата, където ще се изпълнява

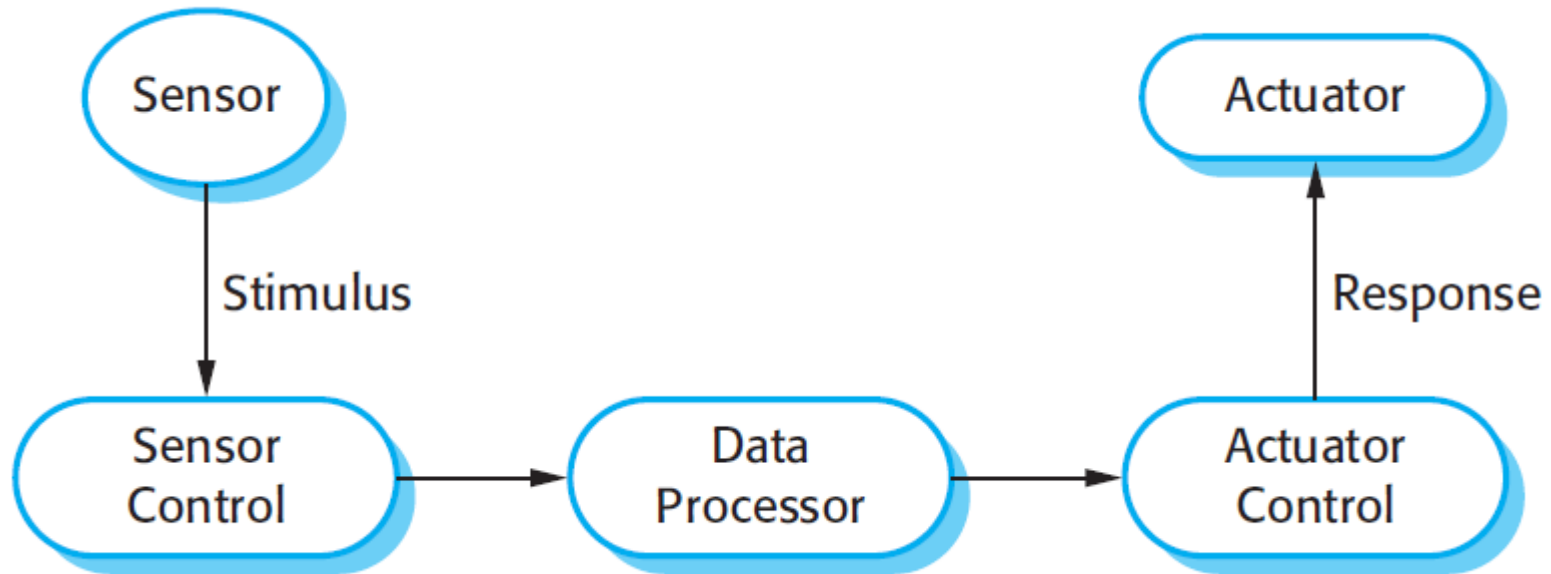
# Характеристики на ВСС

- Обикновено вградените системи работят без прекъсване (няма shutdown)
- Взаимодействието с обкръжението на системата (други системи, хора, машини) непредвидимо
- Може да има физически ограничения, които да влияят на дизайна на системата
  - Напр. мобилните устройства имат изискване за намалена консумация на енергия, което може да доведе до ограничаване на периферните устройства и/или избор на конкретна фамилия процесори
- Понякога се налага директно взаимодействие с хардуера
- Изискванията за надеждност и безопасност влияят на дизайна на системата

# Проектиране на системата

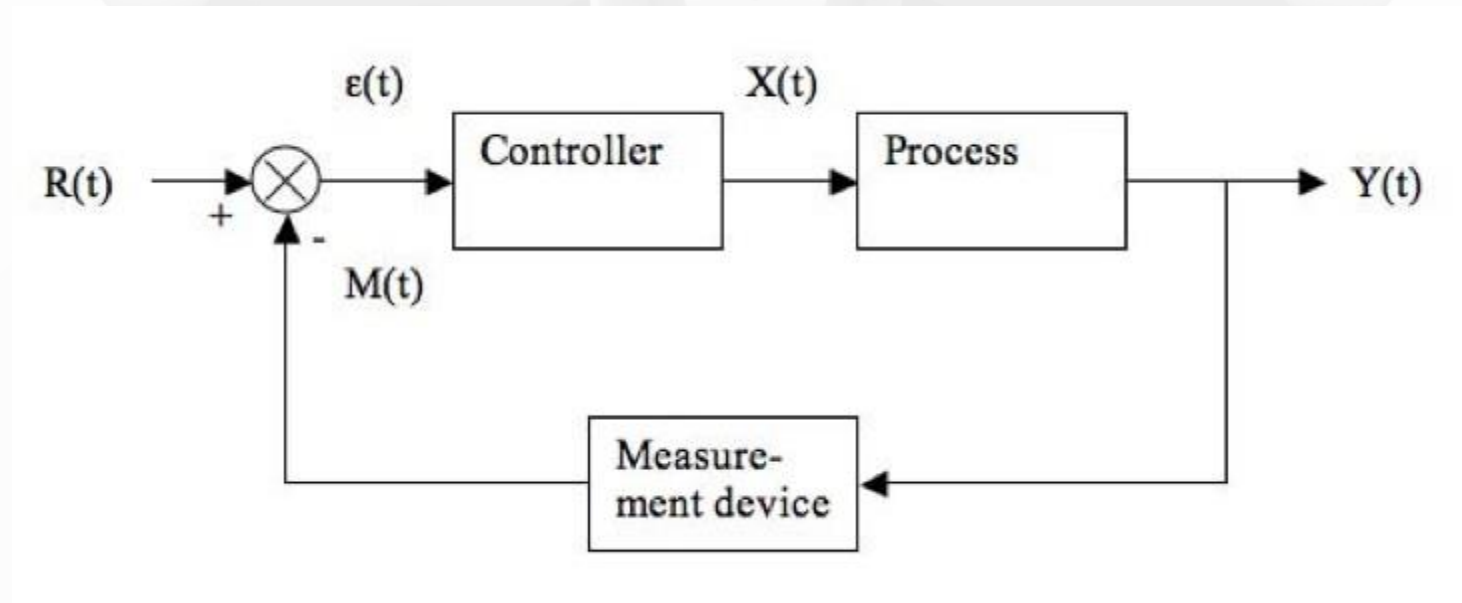
- Едновременно се проектират и апаратното (hardware) и програмното (software) осигуряване
- Решенията по дизайна се взимат на базата на нефункционалните изисквания
- Може да се използва специализиран хардуер
  - По-висока производителност, но по-малки възможности за промяна

# Базова архитектура на вградена система

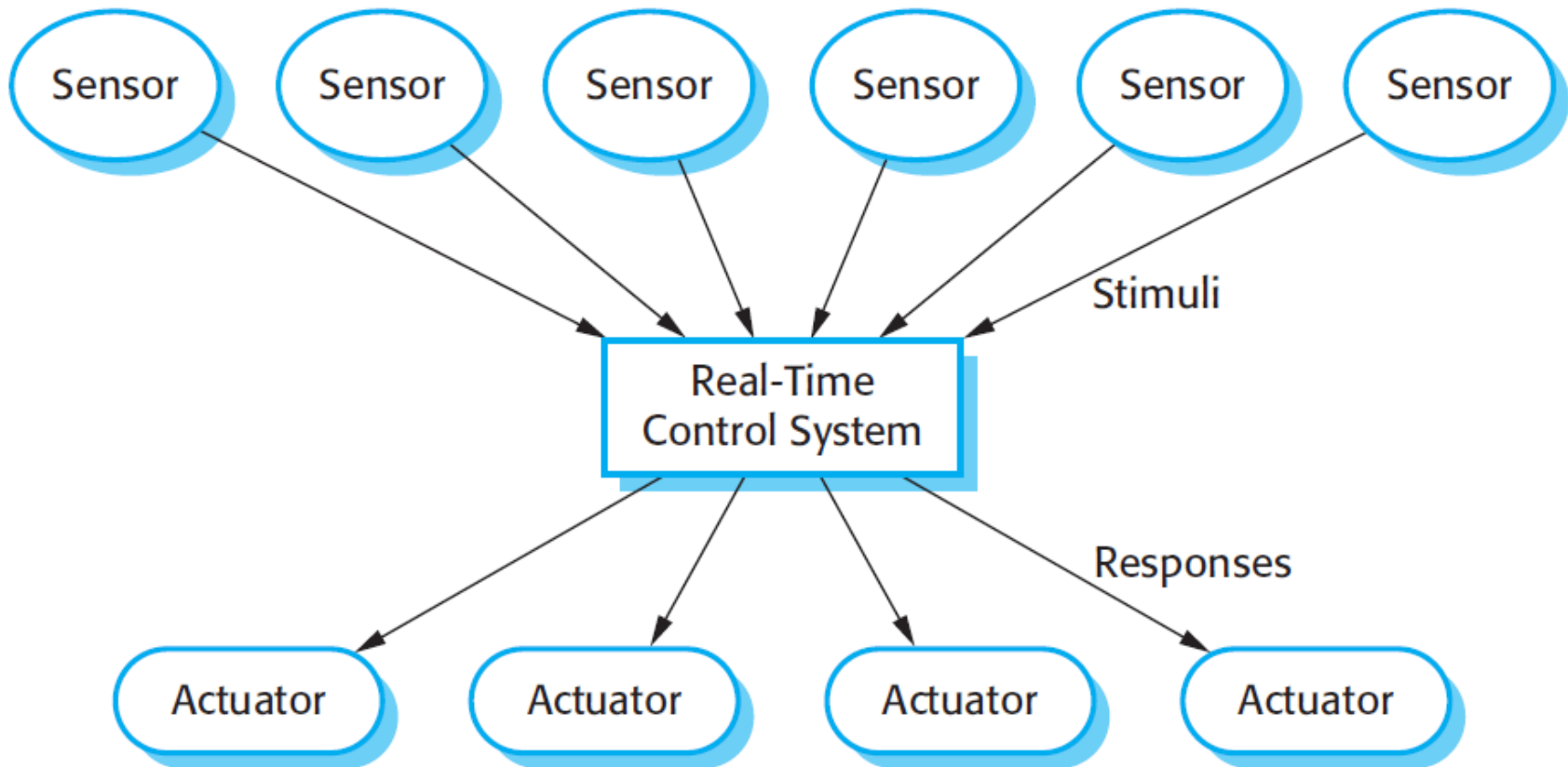


# Класическо (хардуерно) инженерно решение

Принцип на обратната връзка (Control Loop)



# Обобщен модел на система за реално време



# Разработка на вградени софтуерни системи

- Процес за разработка на вградени софтуерни системи
  - Налагат се някои промени в сравнение с традиционните модели на процеси за разработка на системи с общо предназначение

# Процес за разработка на софтуер

Анализ на изискванията  
(Requirements)

Дизайн (design)

Реализация  
(Implementation)

Валидация и верификация  
(Testing, Verification, Validation)

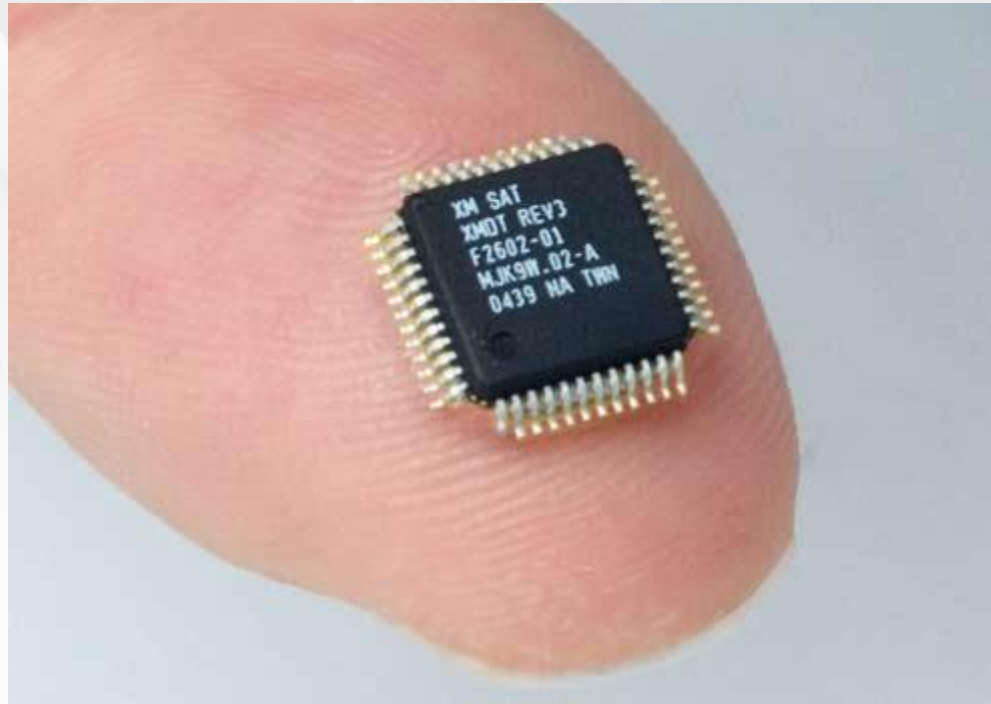
Поддръжка  
(Maintenance)



- Анализ на изискванията
  - Основно значение имат качествените изисквания
- Дизайн
  - Съвместно се проектират хардуера и софтуера
- Разработка на системата
  - Специална развойна среда
  - Езици от ниско ниво (асемблер, ANSI C и др.)
- Тестване
  - Може да се използват симулатори

# Разработка на вградени системи

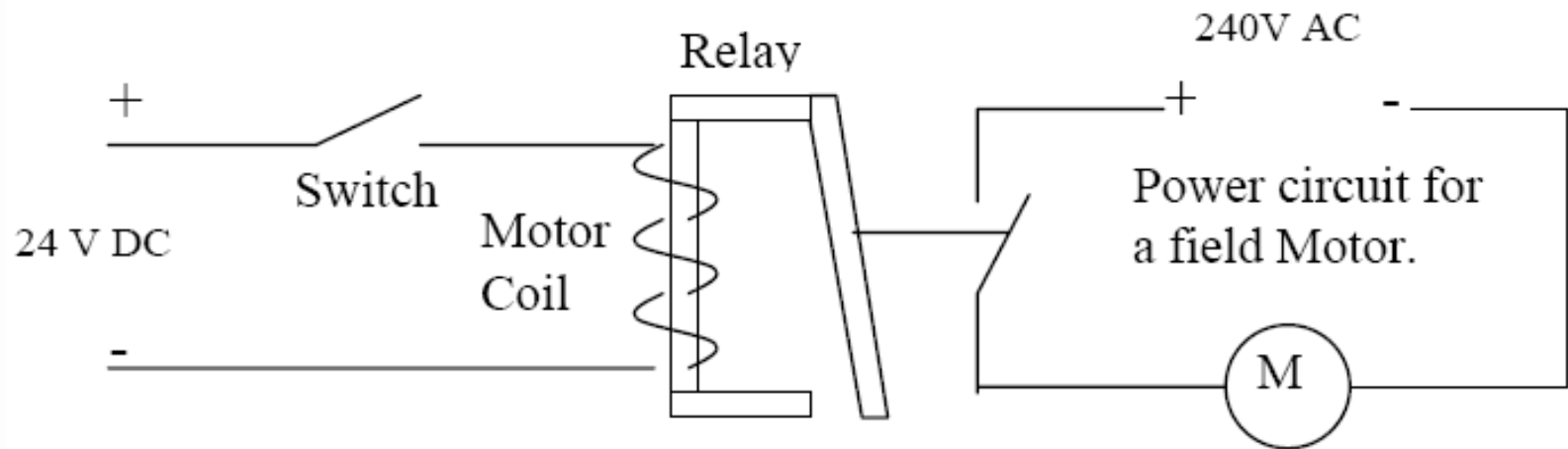
## Едночипов микрокомпютър (Single-chip microcomputer)



# Single-chip microcomputers

- Целият компютър се намира върху единствена интегрална схема (чип)
  - Процесорите
  - Паметта (ROM and RAM)
  - Всички входно-изходни устройства (Video, serial, etc)

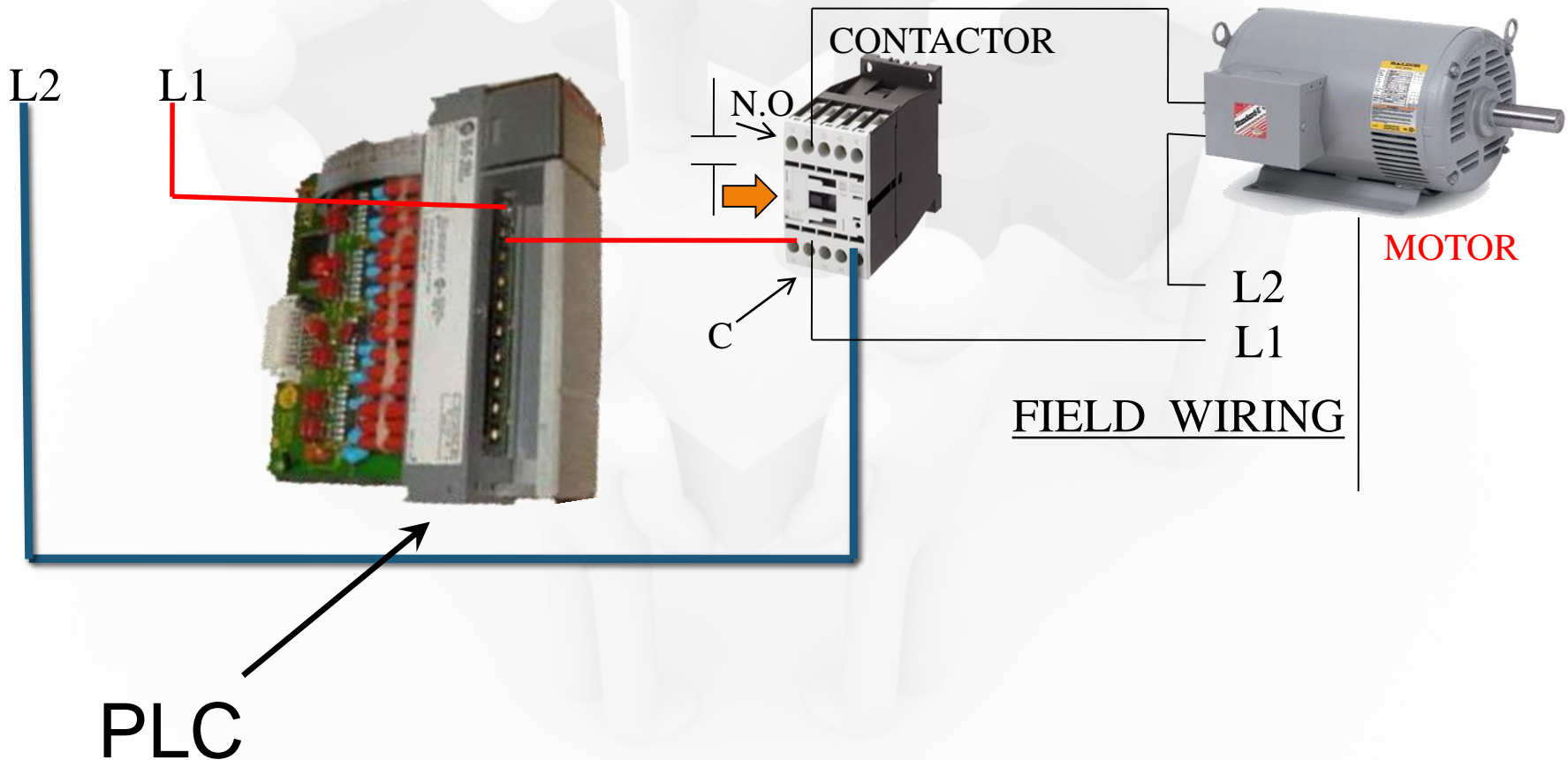
# Системи за управление (до 1980)



# Програмируеми логически контролери (PLC)



# Вградени софтуерни системи за управление



# Разработка на системи за управление

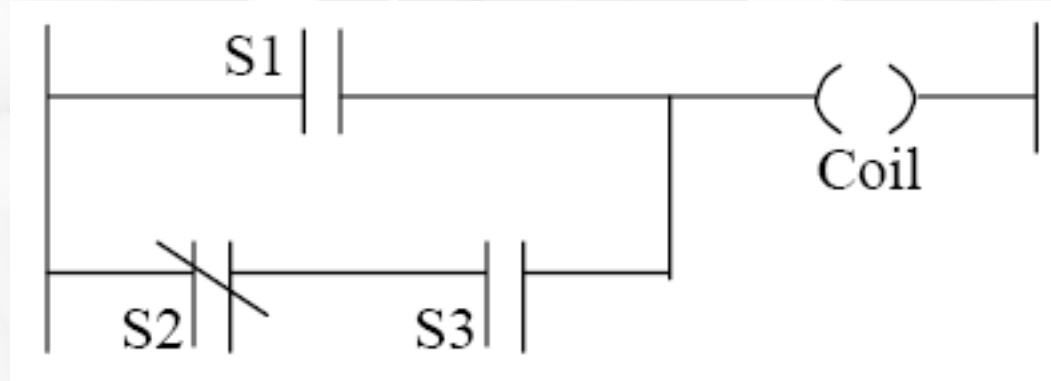
- Програмируеми логически контролери (PLC)
- Стандарт IEC 61131. Включва набор от програмни езици за (PLC)
  - Instruction lists
  - Structured text
  - Ladder diagrams
  - Function block diagrams (FBD)
  - Sequential Function Chart
- Изборът на конкретен програмен език зависи от подготовката на инженера и вида на задачата, която се решава

# IEC 61131-3

- **Instruction List (IL)** - A low level 'assembler like' language that is based on similar instruction list languages found in a wide range of today's PLCs.
- **Structured Text (ST)** - A high level textual language that encourages structured programming. It has a language structure (syntax) that strongly resembles PASCAL and supports a wide range of standard functions and operators. The standard includes a formal syntax definition of ST
- **Function Block Diagram (FBD)** - A graphical language for depicting signal and data flows through function blocks - re-usable software elements. FBD is very useful for expressing the interconnection of control system algorithms and logic.
- **Ladder Diagram (LD)** - A graphical language that is based on the relay ladder logic - a technique commonly used to program current generation PLCs. However, the IEC Ladder Diagram language also allows the connection of user defined function blocks and functions and so can be used in a hierarchical design.
- **Sequential Function Chart (SFC)** - A graphical language for depicting sequential behaviour of a control system. It is used for defining control sequences that are time- and event-driven. This is an extremely effective graphical language for expressing both the high level sequential parts of a control program as well as programming low-level sequences, e.g. to program an interface to a device.

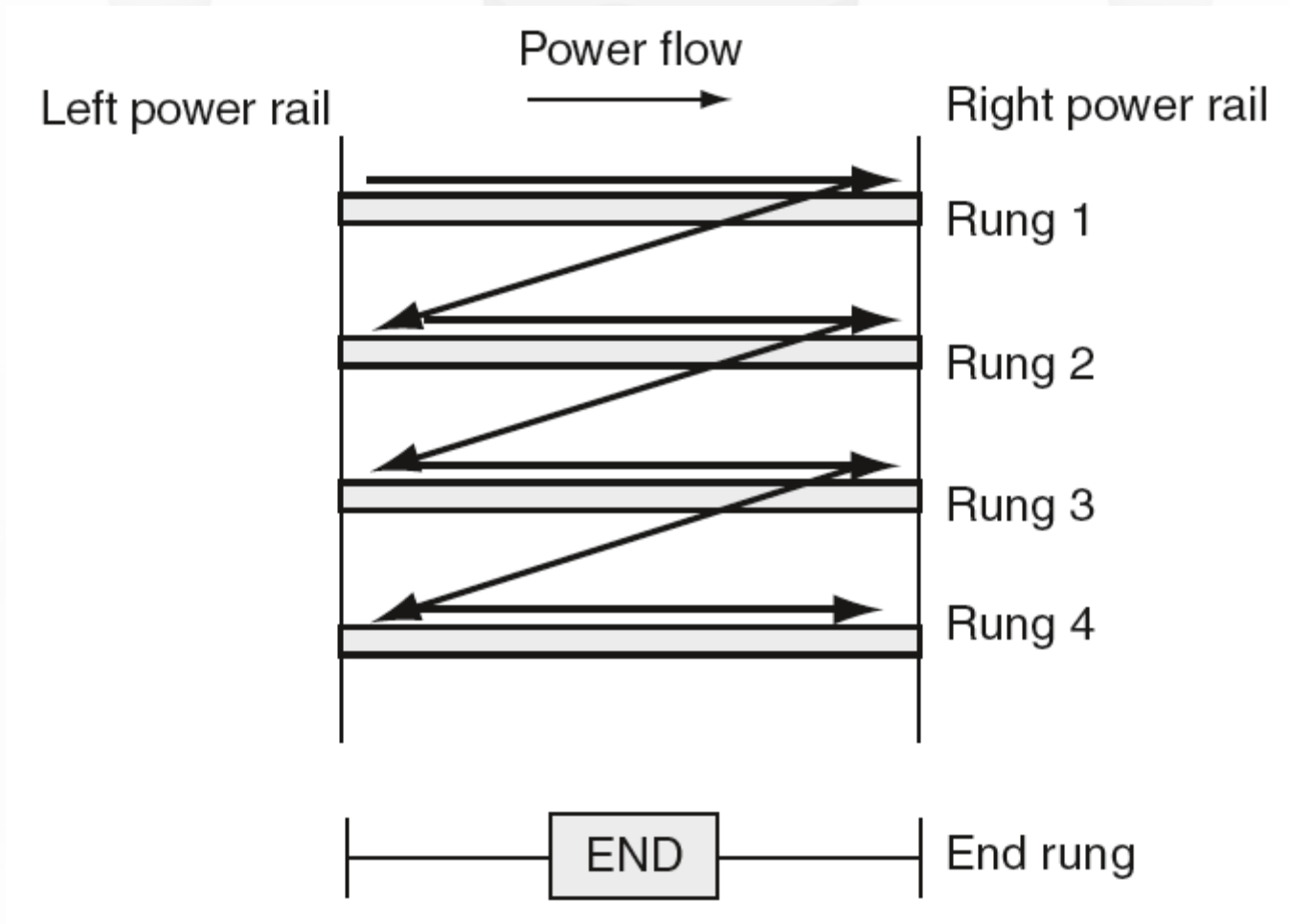


# Ladder diagram

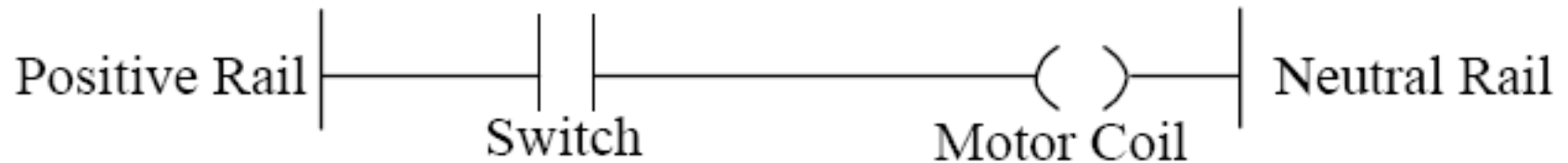


**Coil = S1 OR (!S2 AND S3)**

# Изпълнение на програма на Ladder diagram



- Програмна реализация с Ladder diagrams на схемата с ел. Мотора:

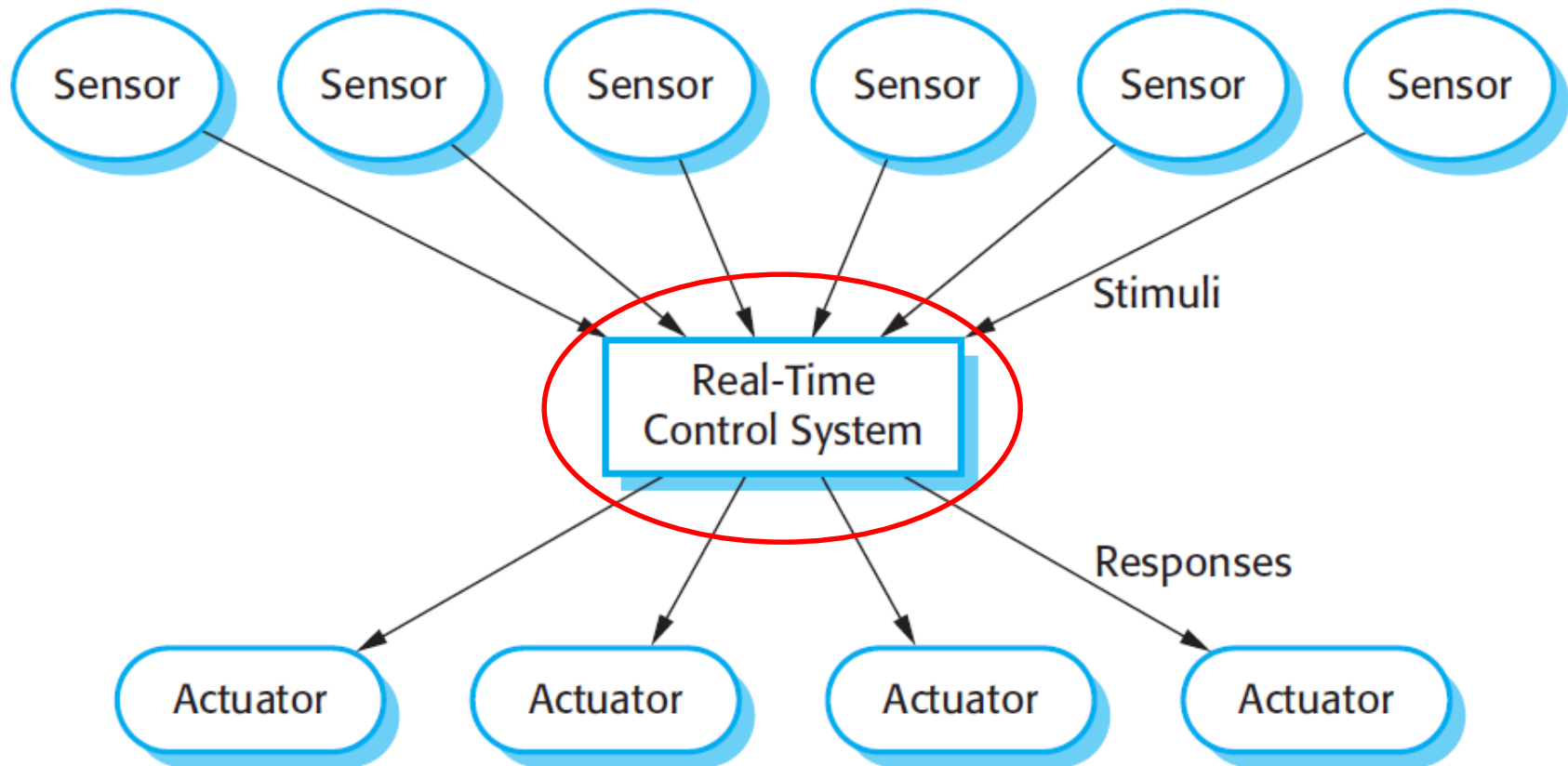


# Системи за реално време

- Дефиниция

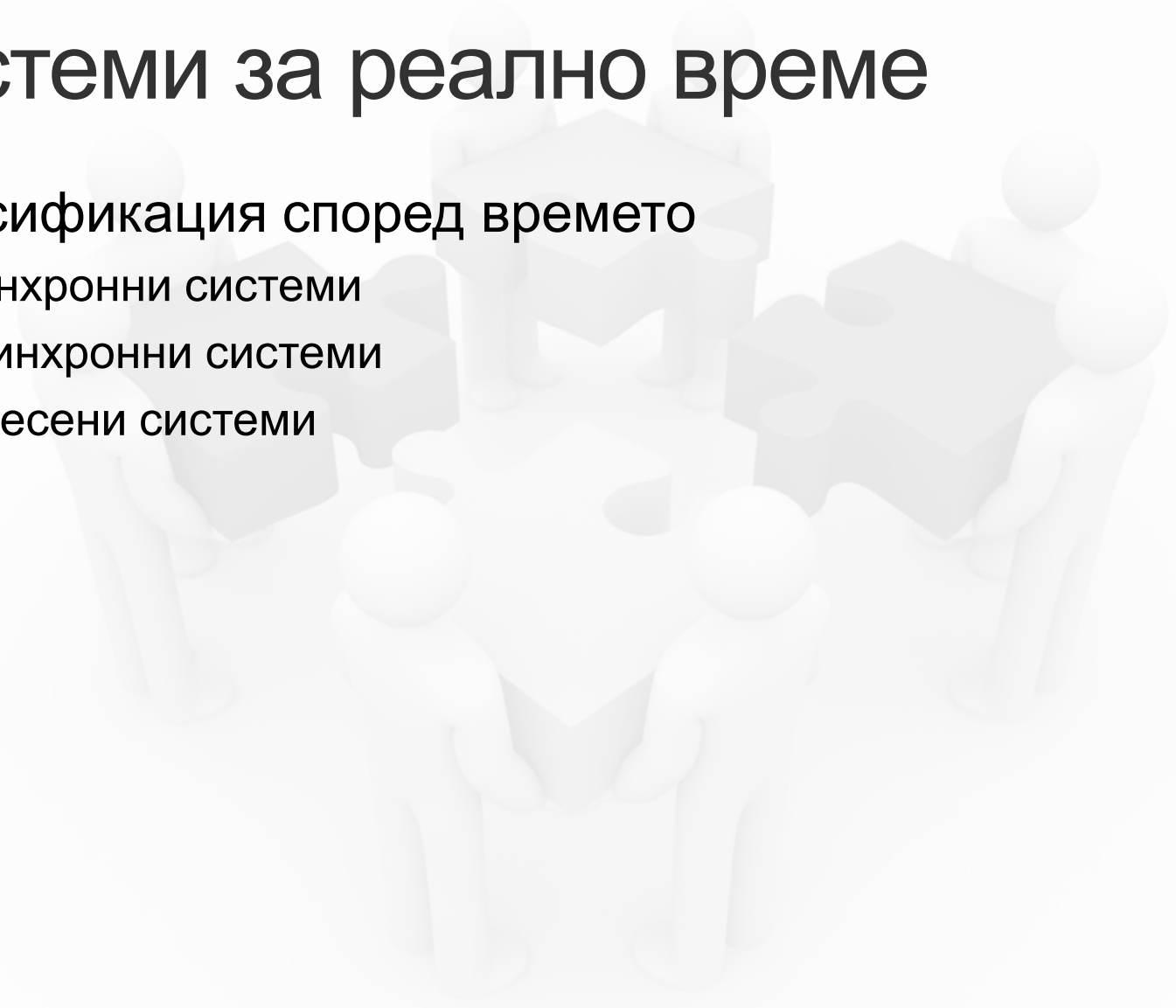
*Система, за която коректността на работата, зависи не само от резултатите, но и от времето за което тези резултати се получават.*

# A general model of an embedded real time system

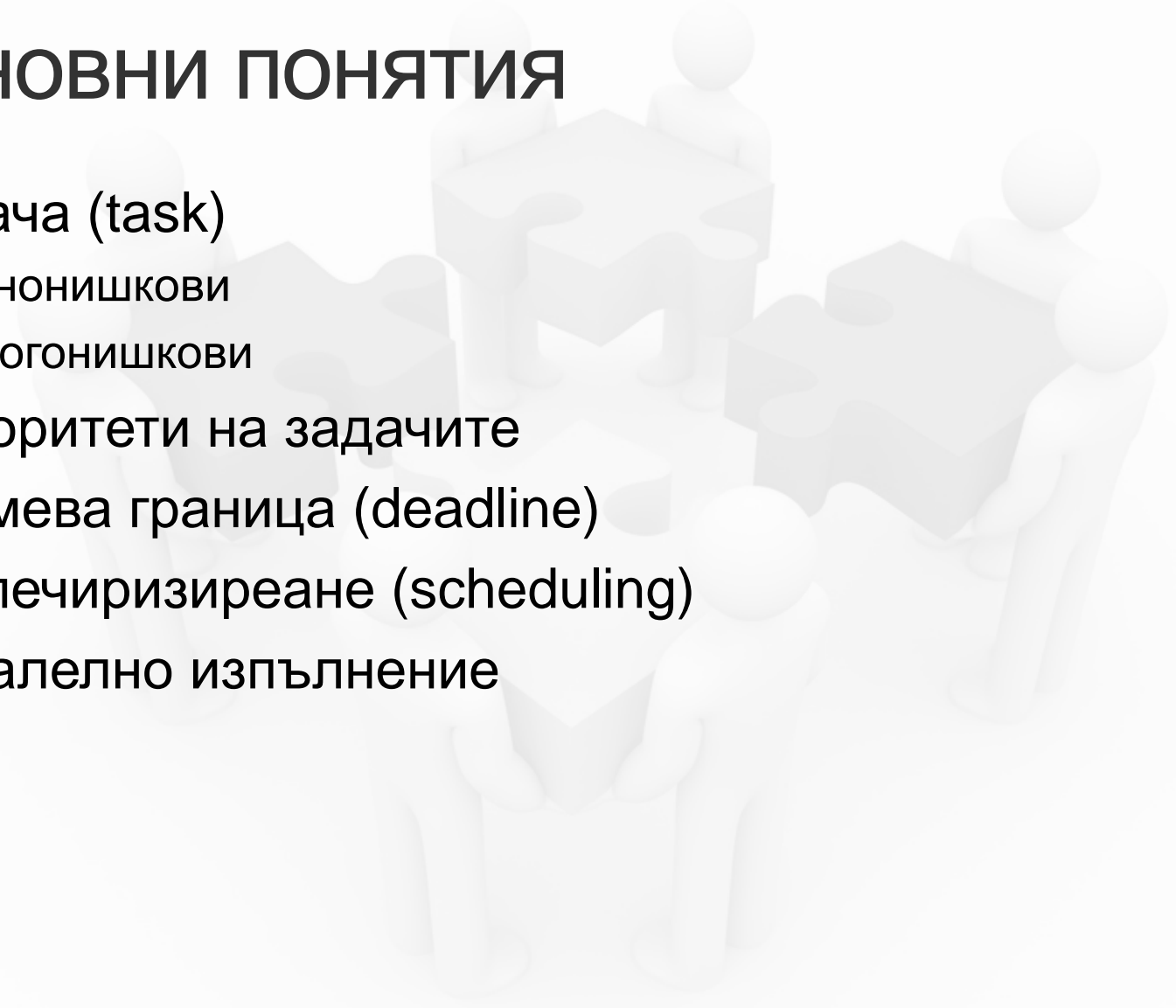


# Системи за реално време

- Класификация според времето
  - Синхронни системи
  - Асинхронни системи
  - Смесени системи



# ОСНОВНИ ПОНЯТИЯ

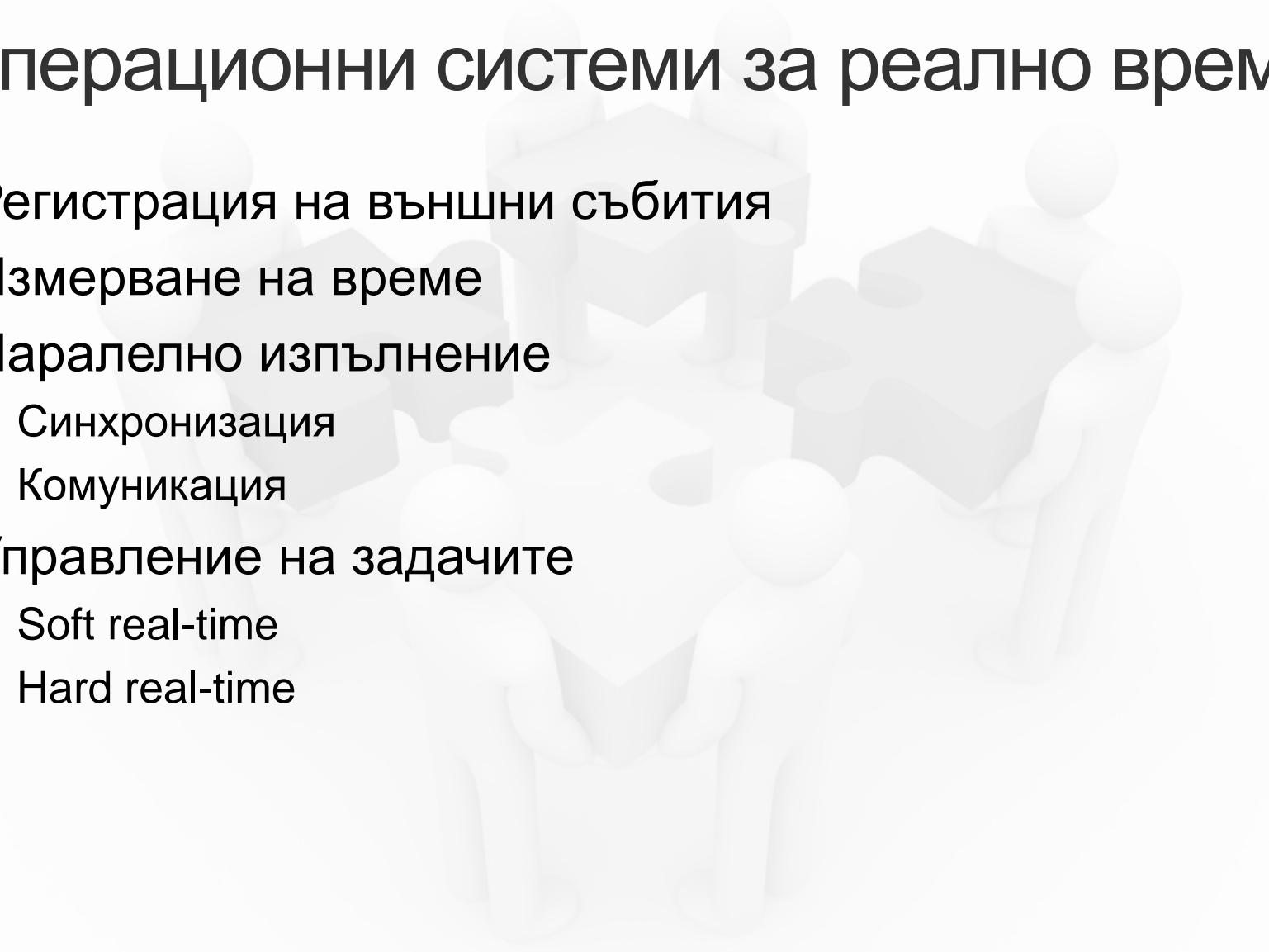
- Задача (task)
    - Еднонишкови
    - Многонишкови
  - Приоритети на задачите
  - Времева граница (deadline)
  - Диспечиризиране (scheduling)
  - Паралелно изпълнение
- 

# Програмни системи за реално време

- Всяка задача трябва да спази времевата си граница
- Област на приложение
  - Критични системи с високи изисквания за сигурност и безопасност
    - Системи в космическите и летателни апарати
    - Системи за управление и др.
- Разчита се на т.нар. операционни системи за реално време (ОСРВ)
  - На английски - Real Time Operating System (RTOS)



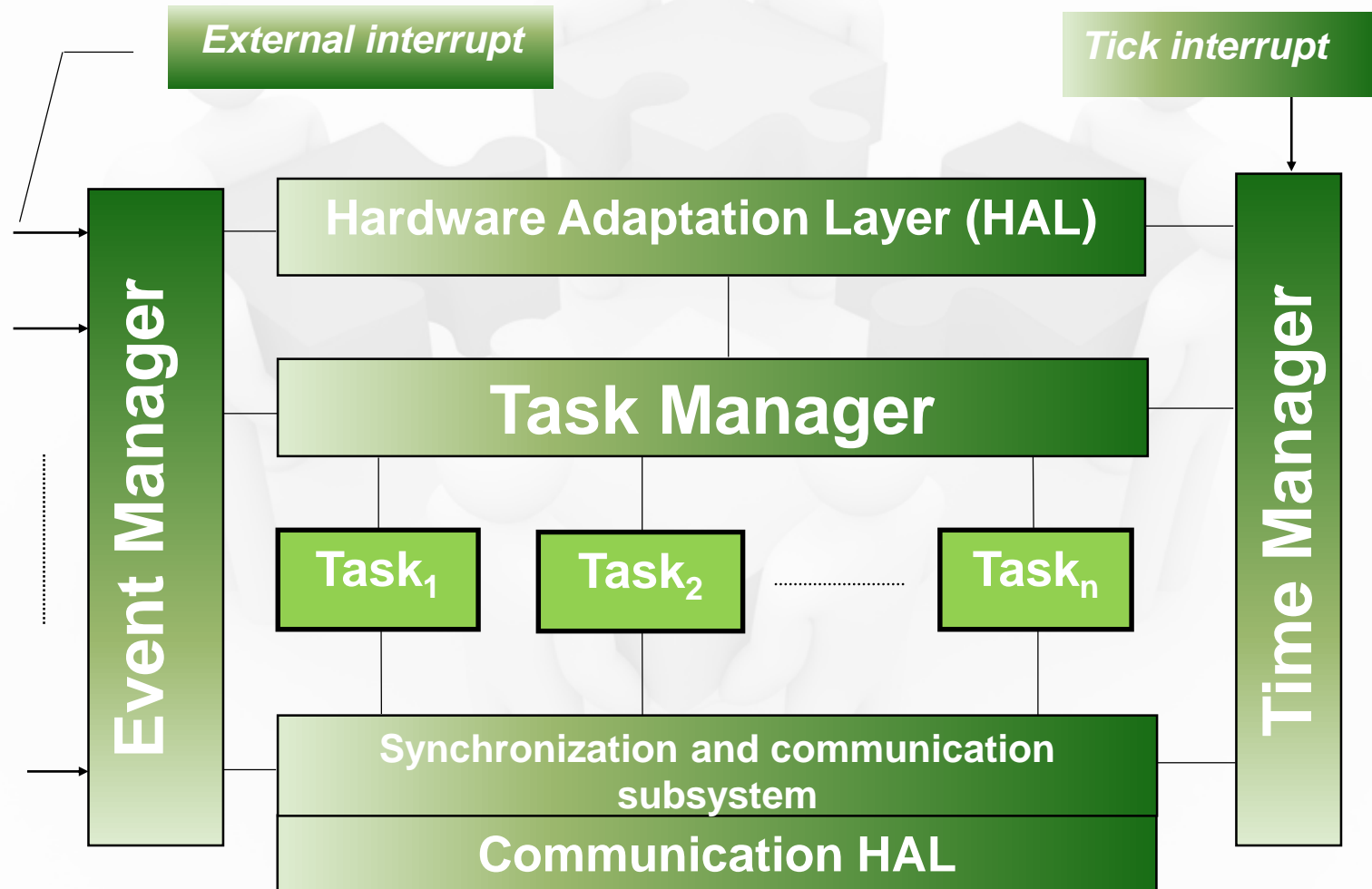
# Операционни системи за реално време

- Регистрация на външни събития
  - Измерване на време
  - Паралелно изпълнение
    - Синхронизация
    - Комуникация
  - Управление на задачите
    - Soft real-time
    - Hard real-time
- 

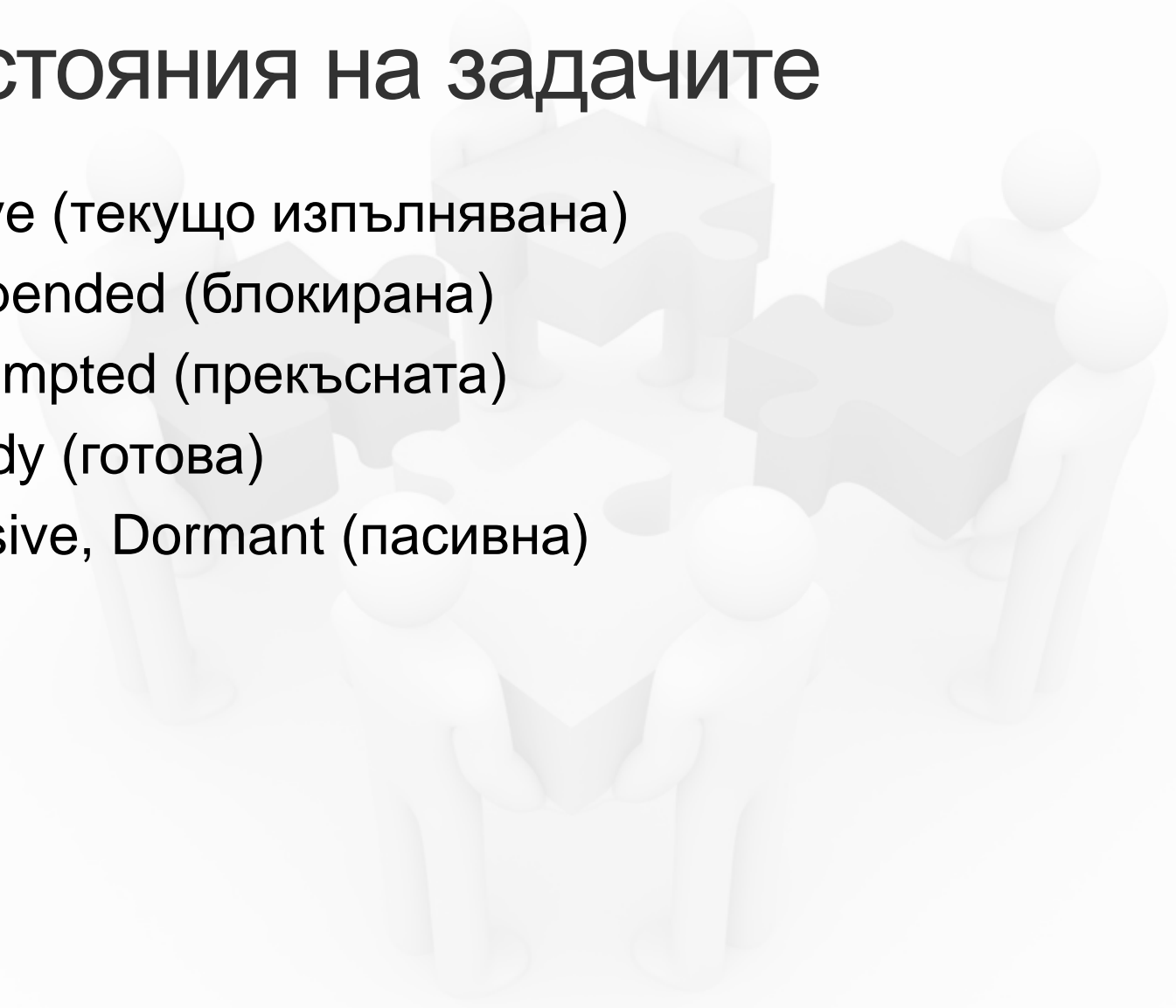
# Операционни системи за реално време

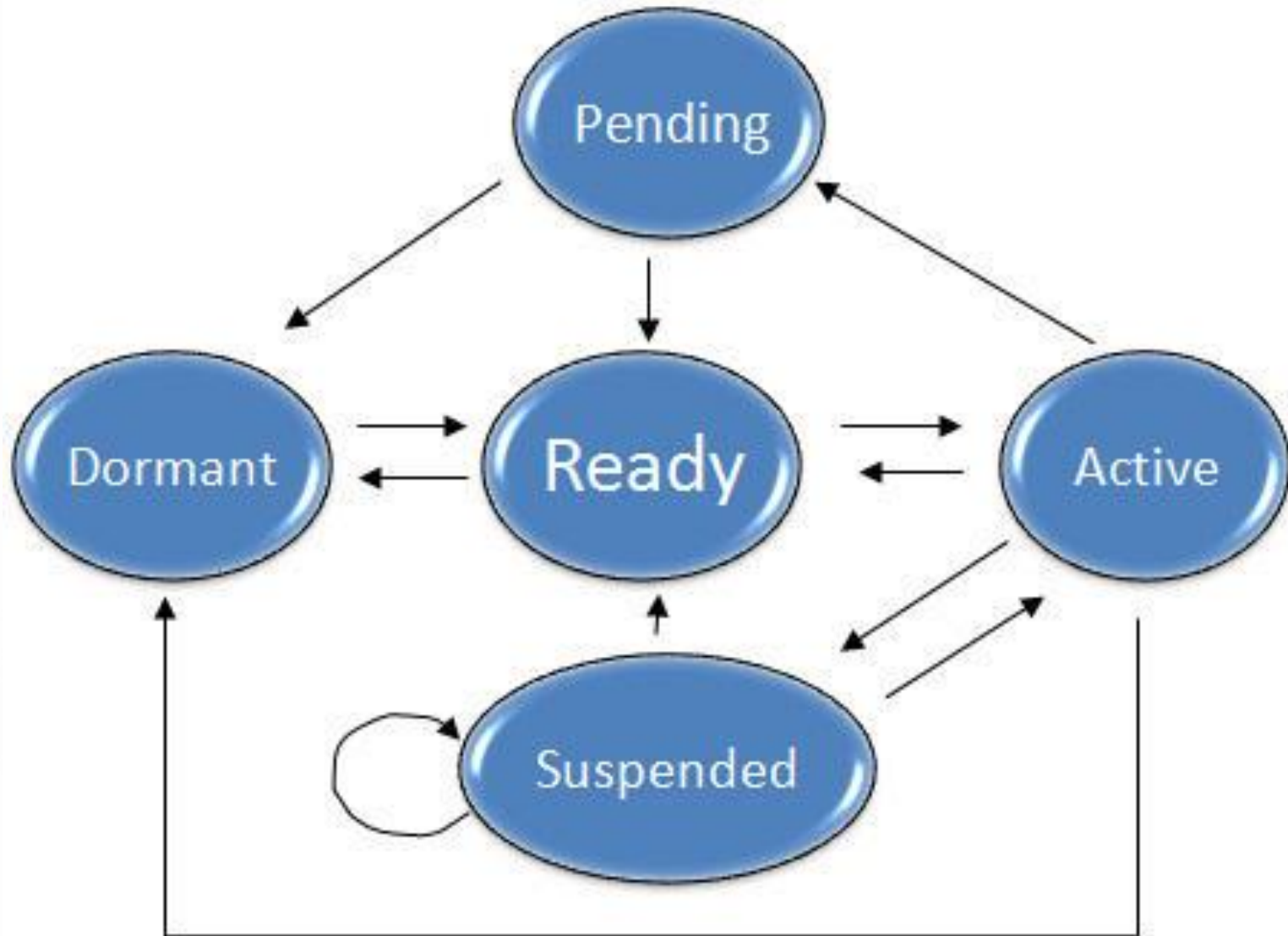
- Класификация според критичността:
  - **Hard Real-Time** - Всеки пропуск на времевата граница може да има катастрофални последствия
  - **Firm Real-Time** - Всеки пропуск на времевата граница води до нежелано намаляване на качеството
  - **Soft Real-Time** - Пропуските във времевите граници са допустими и намаляването на качеството на системата е приемливо
  - **Non Real-Time** - Няма изисквания за спазване на времевите граници

# Примерна архитектура на ОСРВ



# Състояния на задачите

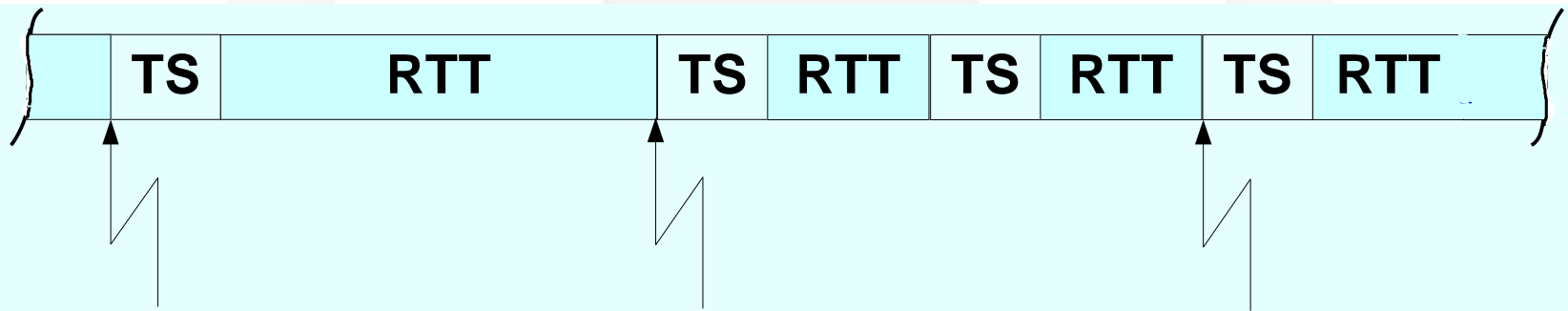
- Active (текущо изпълнявана)
  - Suspended (блокирана)
  - Preempted (прекъсната)
  - Ready (готова)
  - Passive, Dormant (пасивна)
- 



# Task Execution Patterns

- Синхронно изпълнение
  - Понижена реактивност
  - Предсказуемост
  - По-сложна реализация от асинхронните системи
- Асинхронно изпълнение
  - Повишена реактивност
  - По-ниска предсказуемост на системата
  - По-лесна реализация от синхронните системи
- Смесено изпълнение

# Синхронни системи



## Timer IRQ

- RTT – Реално времева задача (Real Time Task)
- TS – Диспечер на задачите (Task Scheduler)

# Асинхронно изпълнение



- ISR – Програма за обработка на прекъсване (Interrupt Service Routine)
  - Може да извиква или да не извиква диспечера на задачите



# Стратегии за диспечиризиране

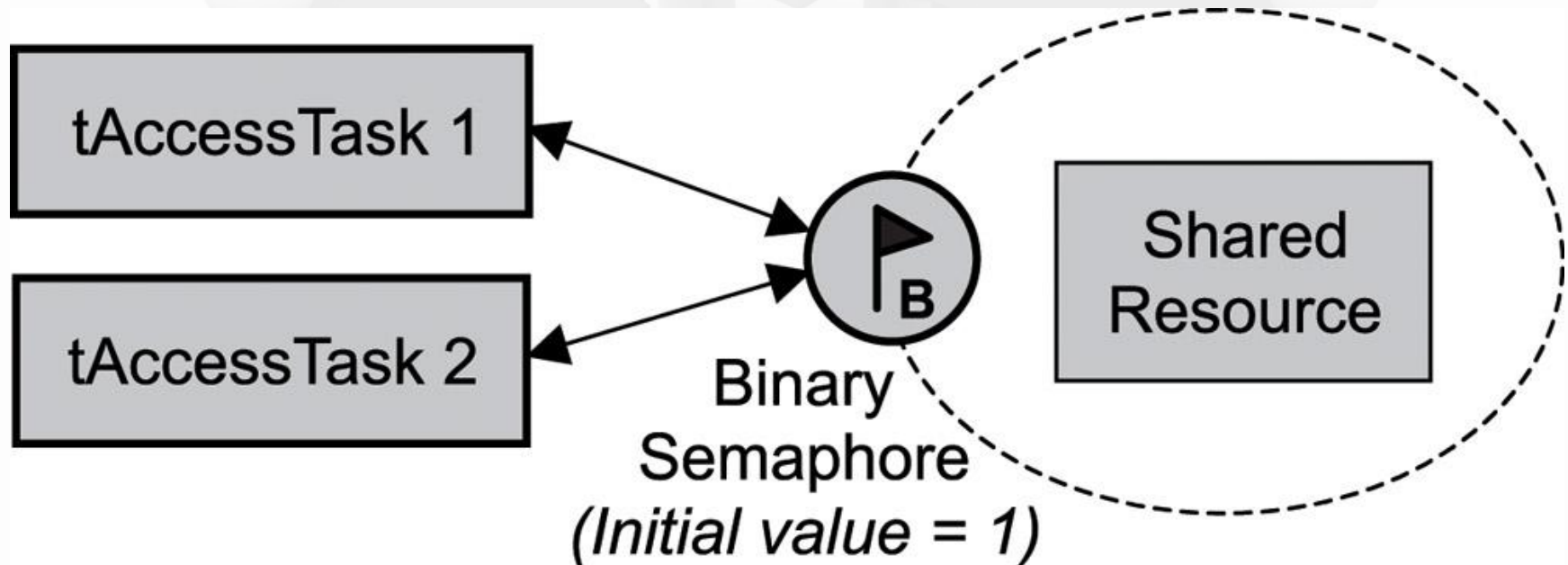
- Не прекъсваемо (Non pre-emptive) диспечиризиране
  - Веднъж след като е старирал, процесът се изпълнява докрай или докато не се блокира по някаква причина (например изчакване за крит. секция)
- Прекъсваемо диспечиризиране (Pre-emptive scheduling)
  - Изпълнението на даден процес може да бъде прекъснато от по-високоприоритетен процес

# Управление на ресурсите

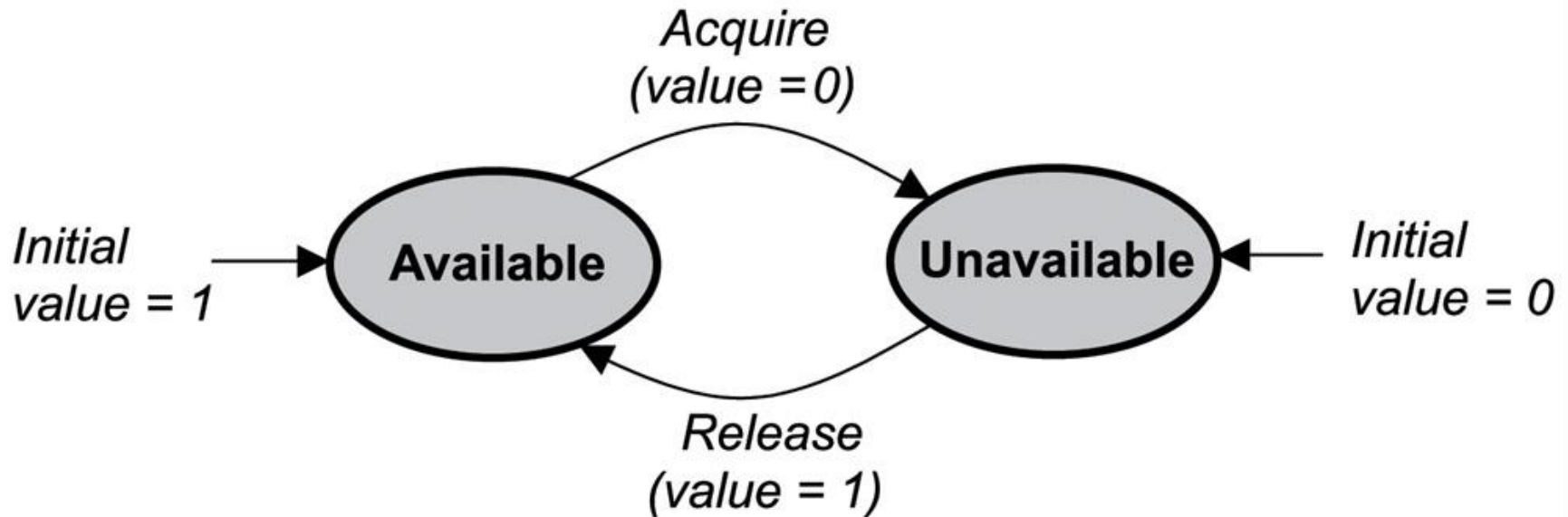
- Критични секции
- Двоични семафори
- Семафори



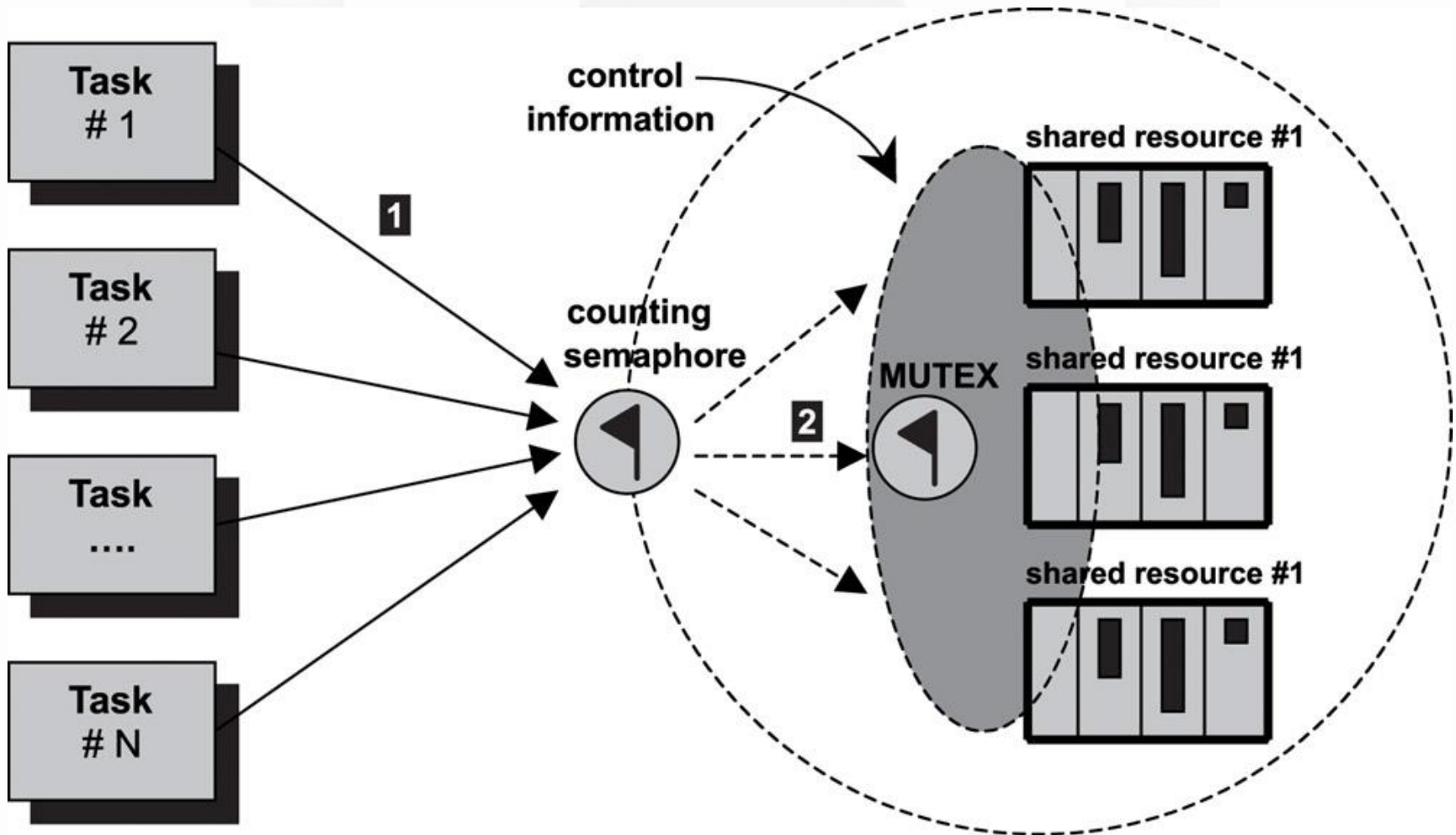
# Двоичен semaфор



# Принцип на работа на семафора



# Семафори



# Процес за проектиране на системи за реално време

- Идентификация на събитията, които трябва да се обработват
- Дефиниране на необходимите реакции на събитията
- За всяко събитие и реакция се определят времевите ограничения
- Обединяване на събитията и реакциите в паралелно изпълнявани процеси

# Процес за проектиране на системи за реално време

- Съставят се алгоритми за обработка на различните видове събития и реакции, които трябва да посрещнат зададените времеви ограничения
- Проектира се система за диспечизиране, която има задача да стартира всеки процес навреме
- Интегрира се всичко чрез ОСРВ

# Времеви ограничения

- Понякога изискват симулации или експерименти за валидация
- Ограничават приложението на някои широко разпространени подходи за проектиране
  - ООД
  - Архитектура, ориентирана към услуги
- От съображения за производителност налагат употребата на програмни езици от ниско ниво