

СИСТЕМИ СУПУТНИКОВОГО ЗВ'ЯЗКУ І НАВІГАЦІЇ

Доповідач – доцент кафедри мереж зв'язку, ХНУРЕ, Україна
Кривенко Станіслав Анатолійович

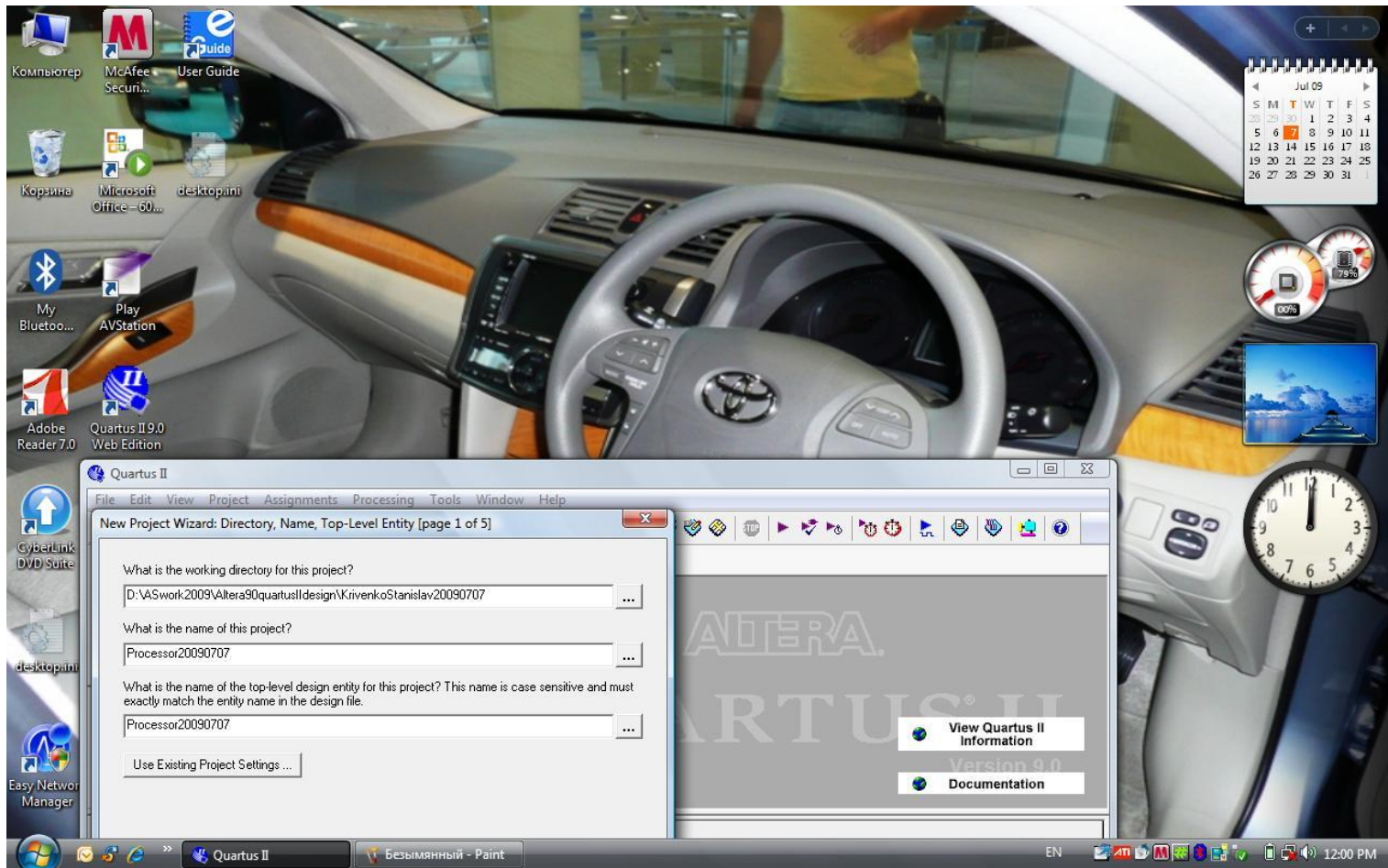
ТЕХНІЧНИЙ ПРОЕКТ ДЛЯ КООПЕРАЦІЇ

2

- **1. НАЗВА ПРОПОЗИЦІЇ**
- **Мікропроцесор із скороченим набором команд для системи супутникового зв'язку і навігації**

Середовище розробки проекту

3



2. СУТЬ ПРОПОЗИЦІЇ

4

- **СУТЬ ПРОПОЗИЦІЇ (короткий і чіткий технічний опис) – ядро мікропроцесора, оформлено у вигляді текстового опису апаратури на мові Verilog (ядро, що синтезується)**

3. ІННОВАЦІЙНІ ХАРАКТЕРИСТИКИ

5

- Процесор технологічно простий, реалізований на мікросхемі корпорації Altera.
- Він дозволяє добитися значної обчислювальної потужності, хорошого часу реакції на переривання, а значить, і часу відгуку системи і т.д.

4. АНАЛІЗ РИНКУ (порівнянні технології або продукти, оцінка)

6

- У сімействі мікросхем є декілька ядер, орієнтованих на різні сегменти ринку: мобільні телефони; GPS-адаптери; маршрутизатори; кишенькові персональні комп'ютери і MP3 програвачі.
- Більшість мікросхем корпорації Samsung побудована на одній і тій же основі (ядро ARM92x різних модифікацій), розрізняючись тактовою частотою, периферією і корпусами. Серія S3C24x0 містить декілька ЦП, основний інтерес представляють S3C24A0A, S3C2410A і S3C2410, які відрізняються типом ядра (ARM926EJ-S і ARM920T відповідно) і наявністю інтегрованого декодера MPEG-4. Вони орієнтовані на ринок традиційних і мультимедійних КПК/смартфонів. S3C24A0A забезпечує достатньо високу продуктивність при обробці сигналів, не в останню чергу завдяки апаратному модулю обробки MPEG-4. Мікросхема має високий ступінь інтеграції, зокрема передбачаються 16-Кбайт кеш інструкцій і даних, MMU, РК контролер, інтерфейс з відео камерою, завантажувальний блок NAND флеш, модуль системного управління (зокрема, забезпечує управління енергоспоживанням), двоканальний UART, 4-канальний контролер DMA, таймери, GPIO, контролер USB-хоста (використовується, наприклад, в Acer, - досить незвичайна для бюджетного КПК можливість), SD/MMC і т.д.
- Модель сімейства S3C2410A користується значно більшою популярністю. Цей ЦП фактично став стандартом для бюджетних КПК, його використовують всі, включаючи HP, Acer, Eten і ін. У корпорації Samsung є ще декілька рішень (наприклад S3C3410), але в КПК вони зустрічаються дуже рідко. З особливо цікавих, варто відзначити мікросхему S3C2410X01, тактова частота якої може досягати 533 МГц при збереженні невисокого енергоспоживання.

5. ПЕРЕВАГИ ПРОПОЗИЦІЇ

7

- **ПЕРЕВАГИ ПРОПОЗИЦІЇ (технологічні, економічні, екологічні, ...)**
- Виникнення в модельних рядах розробників мікросхем тривимірної графіки виробів для кишенькових персональних комп'ютерів і смартфонів обумовлено швидше маркетинговими міркуваннями, ніж реальною потребою ринку. Зважаючи на специфіку мобільних пристроїв, перш за все обмежений за розмірами екран, можна сказати, що 3D насилу прокладає собі дорогу в кишені користувачів. У багатьох випадках при відображенні 3D графіки на екранах мобільних пристроїв не виникає необхідності навіть в банальній фільтрації текстур - із-за невеликої розподільної здатності екрану пікселі просто непомітні. Проте, цей процес, хай і відносно мляво, але йде. І частково щоб застопити місце на ринку, частково для повноти спектру продуктів, всі основні учасники цього ринку наголосилися і тут, у багатьох апаратах також використовуються не дуже цікаві з технічної точки зору, зате надійні і дешеві контролери Epson.

6. СТАДІЯ ПРОЕКТУ

8

- Ідея
- Стадія розробки (НДР)
- Промисловий зразок
- Вже на ринку
- Проведені маркетингові дослідження
- Є бізнес-план
- Макет, дослідний зразок
- ДКР, проектно-кошторисна документація
- Проміжний НІОКР, додаткові дослідження

7. ПРАВА ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ

9

- Є ліцензійна угода
- Партнерські / інші домовленості
- Патент одержаний (у яких країнах)
- Подана заявка на патент (у яких країнах)
- Секретне know-how
- Ексклюзивне право
- Інше

8. ВИД БАЖАНОЇ КООПЕРАЦІЇ

10

- Технічна кооперація
- Комерційна угода з технічним сприянням
- Угода про спільне підприємство
- Ліцензійна угода
- Виробнича угода
- Фінансові ресурси (субпідряд & сумісний підряд)
- Подальші дослідження

9. ВЛАСНІ РЕСУРСИ (технічні і фінансові)

11

- Програмне забезпечення корпорації Microsoft: Windows Vista, Office 2007, Visual Studio 2008
- Система автоматизованого проектування мікросхем QuartusII 9.0, комплект для розробки систем на мікросхемі MAXII.

10. ДОДАТКОВА ІНФОРМАЦІЯ

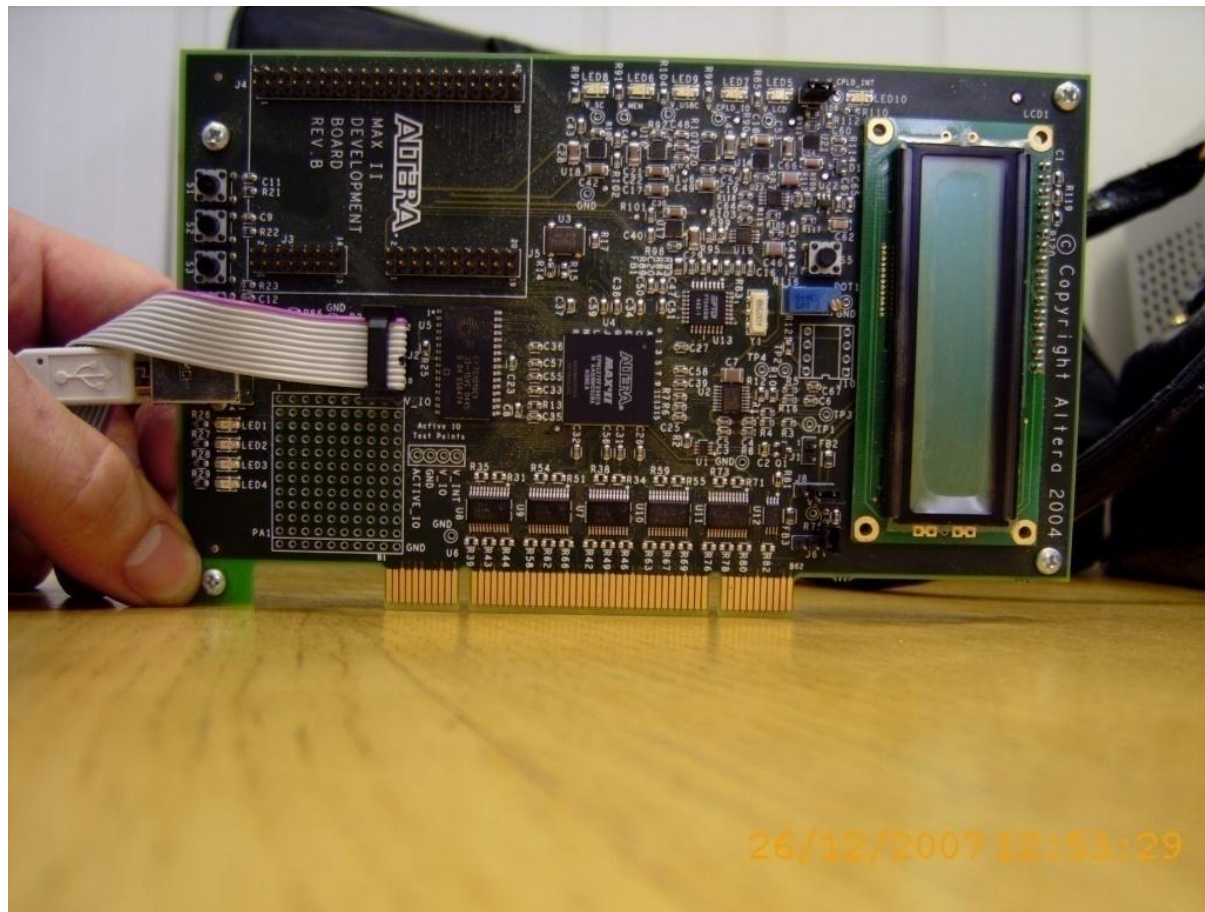
12

- Кафедра “ Мережі зв'язку ”, Харківський національний університет радіоелектроніки, Україна
-
- **Контактна особа**
- Безрук Валерій Михайлович
- Тел./факс: +380577021429
- E-mail: stanisl@vlink.kharkov.ua
- Мова: українська, російська, англійська

11. Пояснювальна записка

13

ПІДКЛЮЧЕННЯ
КОМПЛЕКТУ
КОРПОРАЦІЇ
ALTERA



Операційна система мікропроцесора

- Контекстна схема

Дві форми операційних систем

15

- Взагалі кажучи операційна система може мати дві форми: паралельну та послідовну.
- Кожна з цих форм має свої переваги та недоліки в залежності від застосування в якому вона використовується. Дійсно, перевага однієї з систем має тенденцію бути недоліком іншої.

Послідовна операційна система

16

- Послідовна операційна система заснована на розподілі часу мікропроцесора на послідовність інтервалів.
- На будь-якому інтервалі явно визначено які задачі можуть виконуватися або звертатися до мікропроцесора.
- Як тільки програмна задача активована, вона утримує мікропроцесор до свого завершення, а операційній системі немає потреби зберігати контекст програмної задачі та мікропроцесора

Виключення в послідовній операційній системі

17

- Виключення можливе в системі, коли нормальна програмна операція зазнає дію фізичного переривання, в цьому випадку необхідно зберегти контекст, виконати обробку переривання та відновити контекст.
- Внаслідок згаданого вище послідовна операційна система застосовує мінімум системних ресурсів: простір пам'яті та мікропроцесор, який завантажується зверху донизу.
- Основний недолік системи – це її нездатність працювати в наступних системах: реального часу, де необхідно виконувати множину задач з широким діапазоном часу виконання та де задачі мають різні пріоритети.

Паралельна операційна система

18

- Паралельна операційна система призначена для підтримки набору програмних задач, які виконуються паралельним методом. В цій системі кожній задачі надається час мікропроцесора, а коли виникає конфлікт задач, виконується задача з найбільшим пріоритетом.
- Паралельна операційна система є системою реального часу. Для реалізації концепції паралельного перемикання задач операційна система повинна зберігати контекст програмної задачі та мікропроцесора, а потім поновлювати їх перед тим як поновити виконання задачі.
- Паралельні операційні системи мають велику гнучкість для модифікації існуючих та доданню нових задач в програмний проект.

Послідовна операційна система

19

| Переваги | Недоліки |
|--|--|
| Проста структура для послідовних задач | Не влаштовує виконання операцій у реальному часі |
| Легше виконувати завантаження мікропроцесора | Не влаштовує задачі з різними пріоритетами |
| Мінімальні вимоги до пам'яті для збереження контекстів задач | Відсутня гнучкість при внесенні змін в програмну структуру |
| Легка перевантаження мікропроцесора | Не дозволяє легко вирішувати призначення пріоритетів |
| -- | Важко повно використовувати потужність процесору |

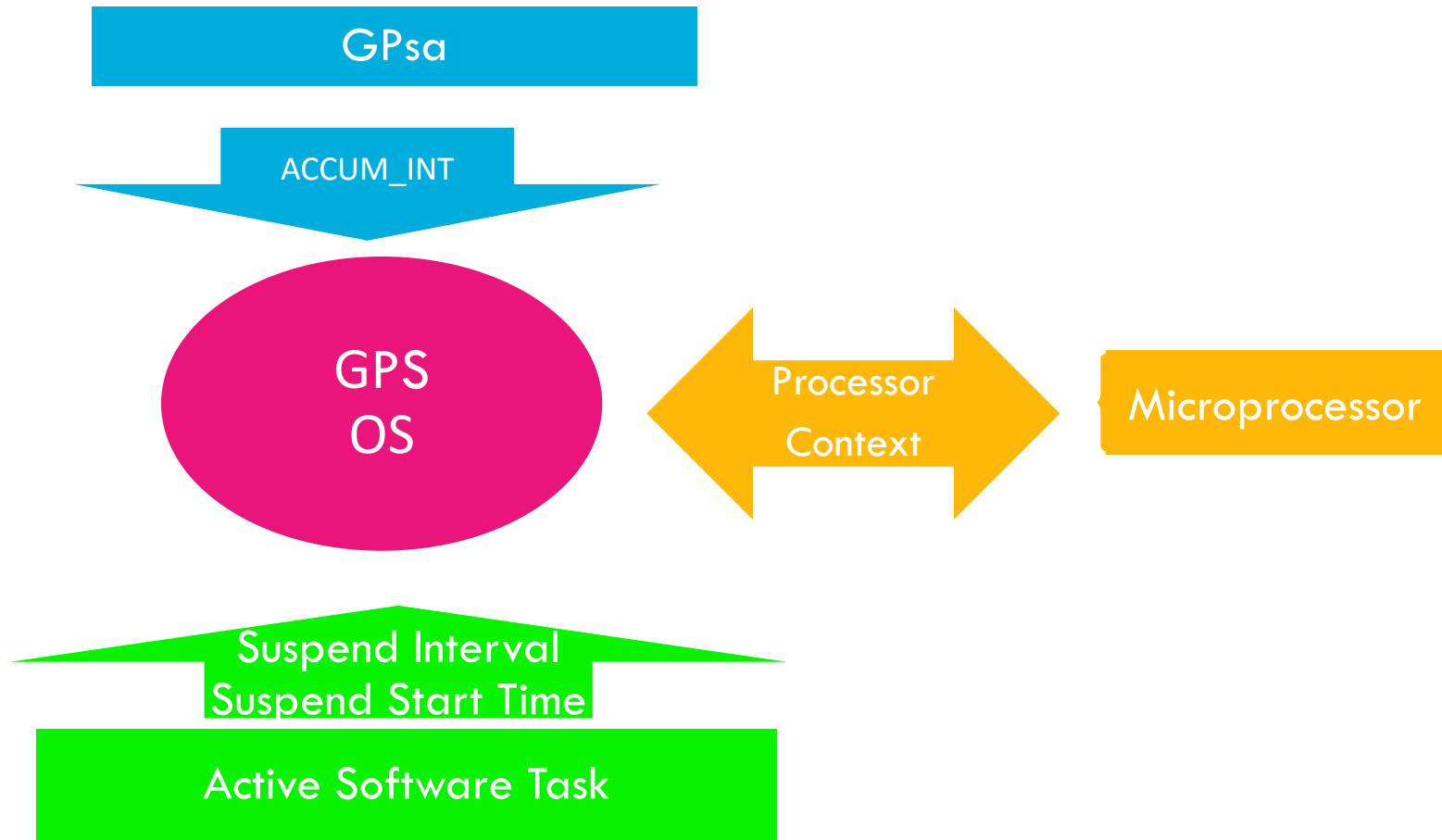
Паралельна операційна система

20

| Переваги | Недоліки |
|--|---|
| Пасує до операцій у реальному часі | Необхідна пам'ять великої місткості для збереження контексту задачі |
| Пасує до задач з різними пріоритетами | Додаткове завантаження мікропроцесора для перемикання задач |
| Вирішує задачу надання пріоритетів | Велика та складна структура |
| Гнучка до додавання нових програмних задач | Важко передбачити навантаження мікропроцесора |

Контекстна схема операційної системи GPS

21



Система управляється перериванням ACCUM_INT GP_{sa}

22

- В першу чергу система управляється перериванням ACCUM_INT GP.
- Це переривання використовується як механізм для знаходження маркерів часу TIC(0.1с), модифікації лічильника цих маркерів та зміни станів програмних задач: активний чи тимчасово зупинений.
- Інший стимул для операційної системи надходить від самозупинки програмної задачі після завершення всіх програмних операцій задачі.

Ініціалізація задач

23

- Перед тим як операційна система GPS почне нормальну роботу необхідно виконати ініціалізацію для завдання конфігурації управляючого блоку задач (Task Control Block – TCB) та контекст кожної задачі. Схеми потоків для цих операцій наведені на рис.
- При ініціалізації кожної задачі виконуються наступні операції: розподіл простору пам'яті під стек задачі, завдання покажчика вершини стеку та ліміту стеку задачі, а також збереження контексту мікропроцесору в стек задачі.
- Пріоритет задачі визначається номером задачі в масиві TCB структур. Найбільший пріоритет має перша задача, а самий низький пріоритет має остання задача, якою завжди виступає задача MAIN.

Контекст задачі

- Оригінальний контекст задачі, який завантажується до мікропроцесора при першій активації задачі.
- Контекст задачі складається з 16 основних реєстрів (r0-r15) та реєстра стану поточної програми CPSR для даної задачі. Тільки наступні реєстри вимагають вірної ініціалізації:
 - R10(sl) – реєстр ліміту стеку задачі(застосовується для перевірки переповнення стека);
 - R13(sp) – покажчик вершини стеку (оригінальне значення база стеку плюс 0x44);
 - R15(PC) – лічильник задачі(оригінальна точка входу в задачу);
 - CPSR – реєстр статусу поточної задачі.

Оригінальне визначення стеку

25

| Показчик стеку | Регістр | Значення |
|-------------------|----------|---|
| База стеку | r0..r9 | Don't care |
| База стеку – 0x28 | r10(sl) | Ліміт стеку задачі |
| База стеку – 0x2C | R11, r12 | Don't care |
| База стеку – 0x34 | R13(sp) | Показчик стеку задачі (оригінальне значення „ База стеку + 0x44) |
| База стеку – 0x38 | R14 | Don't care |
| База стеку – 0x3C | R15(PC) | Точка входу в задачу |
| База стеку – 0x40 | CPSR | Регістр статусу поточної програми |

Схема потоків даних при ініціалізації задачі (Parent: GPS Architect OS)

26

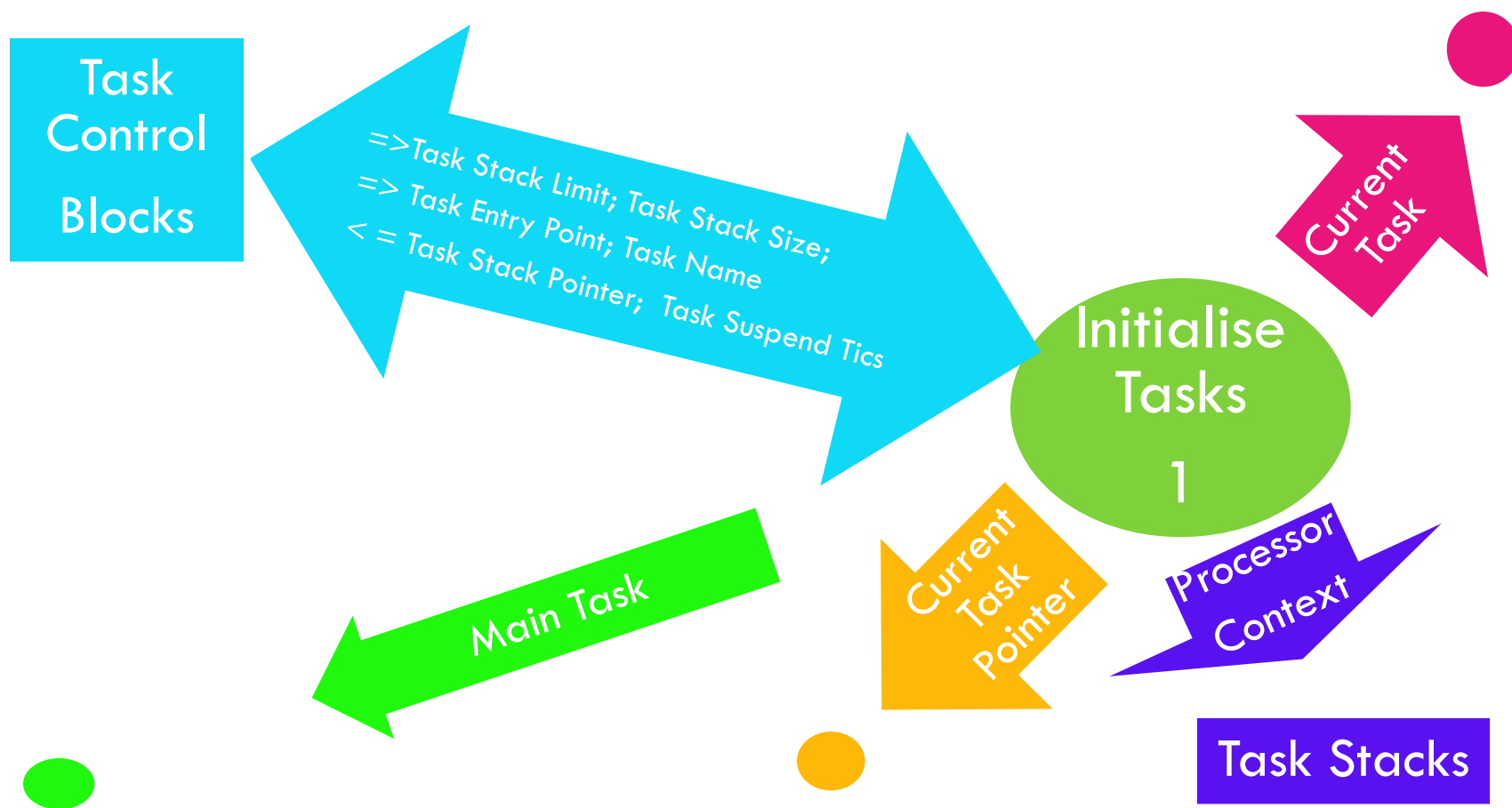
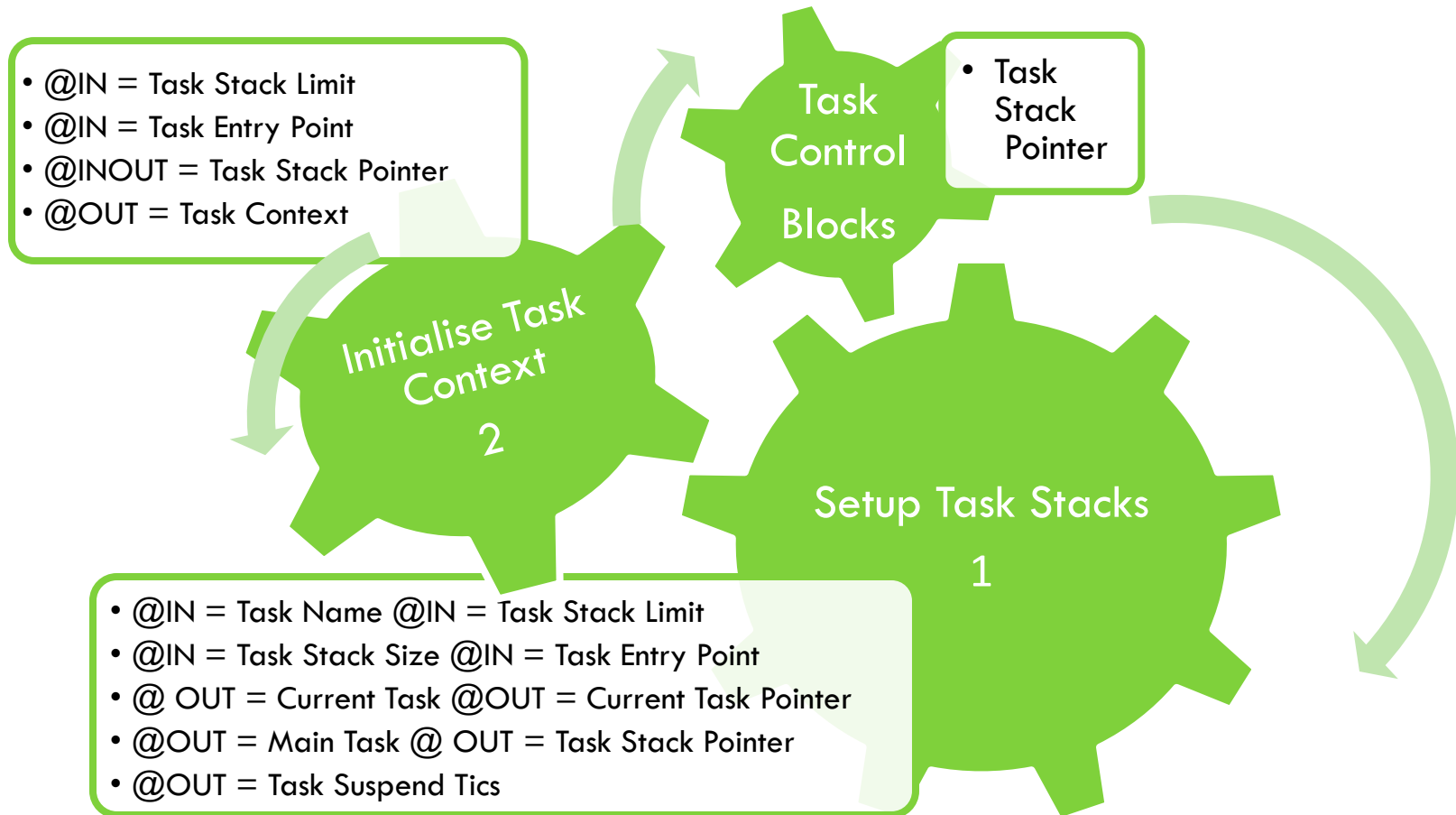


Схема потоків даних при формуванні стеків та ініціалізації контексту задачі (Parent: Initialise Tasks)

27



Специфікація процесу завдання стеків програмної задачі

28

- @IN = Task Name @IN = Task Stack Limit @IN = Task Stack Size @IN = Task Entry Point @OUT = Current Task @OUT = Current Task Pointer @OUT = Main Task @OUT = Task Stack Pointer @OUT = Task Suspend Tics
- @PSPEC Setup Task Stacks
- //For each task, (except MAIN), get a pointer to the stack limit and stack pointer
- task counter = 0; DO Task Stack Limit = memory allocation (Task Stack Size);
- Task Stack Pointer = Task Stack Limit + Task Stack
- task counter = task counter + 1
- WHILE (Task Name != "MAIN")
- // Set the current task and current task pointer to be MAIN and MAIN TCB
- Current Task = Main Task = task counter;
- Current Task Pointer = Task Entry Point of Main Task
- MainTaskSuspendTics = 0 FOR task counter 0 TO MainTask -1 TaskSuspendTics = 0 Initialise Task Context() ENDFOR
- @

Специфікація процесу ініціалізації контексту програмної задачі

29

- @IN = Task Stack Limit @IN = Task Entry Point
- @INOUT = Task Stack Pointer
- @OUT = Task Context
- @PSPEC Initialize Task Context
- Save Task Context to Task Stack Pointer in descending stack:R0-r15,CPSR
- Where: R10 = Task Stack Limit R13 = Task Stack Pointer R15 = Task Entry Point Others registers = don't care
- Update Task Stack Pointer with new stack pointer
- @

Обробка переривань

- Програма обробки переривань (Interrupt Service Routine – ISR) зберігає контекст поточної задачі, якою управляє мікропроцесор.
- Коли надходить ТІС зменшуються на один інтервали тимчасової зупинки задач та поновлюється виконання задачі з найвищим пріоритетом.
- Коли ТІС не надходить поновлюється задача, яка була збережена до входу в ISR.
- Коли ТІС надійшов перемикач задачі може змінитися.

Схема потоків даних при обробці переривань (Parent: GSM Architect OS)

31

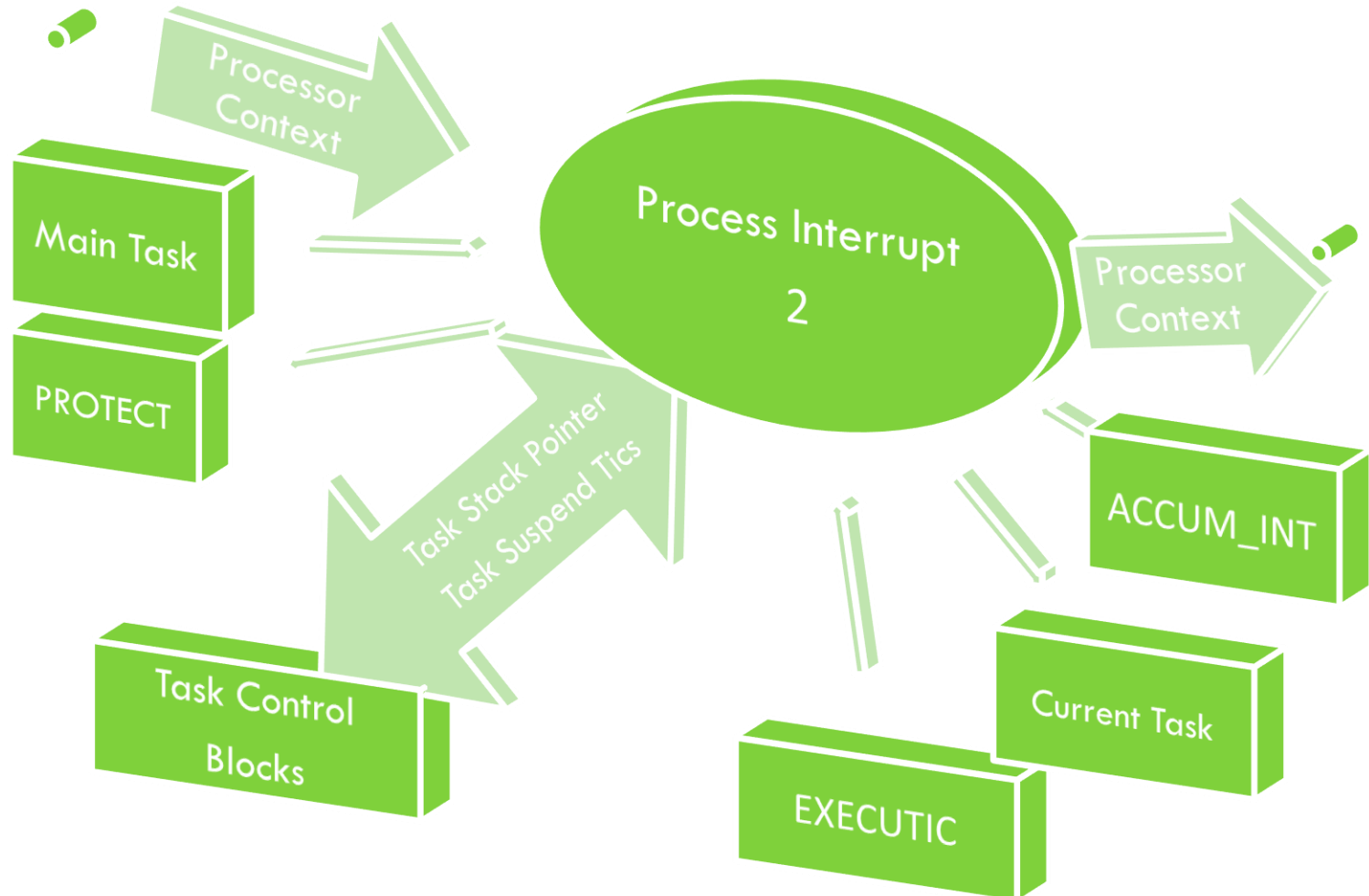
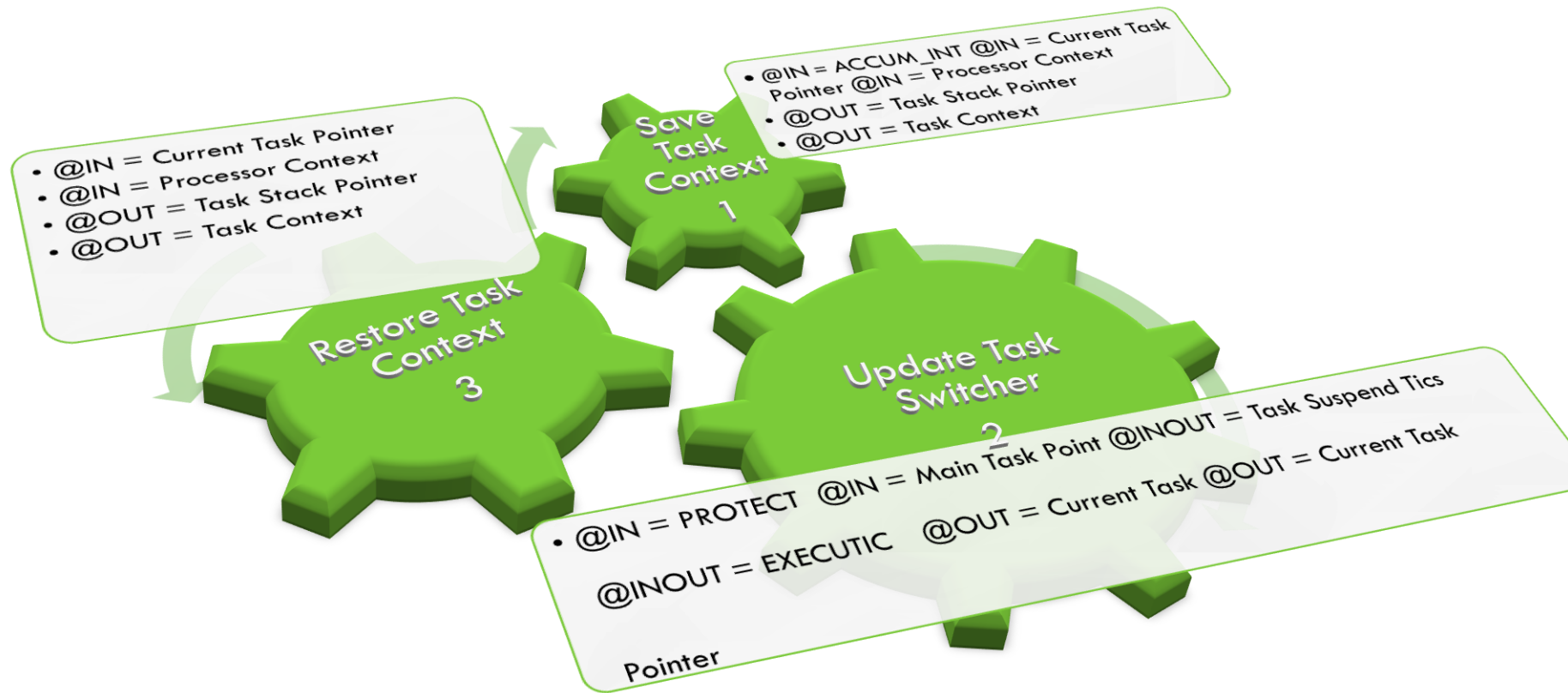


Схема потоків даних (Parent: Process Interrupt)

32



Специфікація процесу збереження контексту програмної задачі

33

- @IN = ACCUM_INT
- @IN = Current Task Pointer
- @IN = Processor Context
- @OUT = Task Stack Pointer
- @OUT = Task Context
- @PSPEC Save Task Context
- Upon an ACCUM_INT(the microprocessor is in IRQ mode).
- Store the User Mode Processor Context (the interrupt task Task Context)
- R0-r14, SPSR&pc
- to the stack of the interrupted task.
- Store the User Mode Stack Pointer
- (the new Task Stack Pointer of the interrupted task)
- to the TCB of the interrupted task (pointed to by Current Task Pointer)
- @

Специфікація процесу зміни перемикача програмної задачі

34

- @IN = PROTECT @IN = Main Task Point
- @INOUT = Task Suspend Tics @INOUT = EXECUTIC
- @OUT = Current Task @OUT = Current Task Pointer
- @PSPEC Update Task Switcher
- //If task switching is allowed and TICs have occurred.
- IF PROTECT==0 and EXECUTIC!=0
- //Decrement the suspend intervals for all tasks.
- FOR Task = 0 TO Main Task -1 IF Suspend Tics for task > EXECUTIC Task Suspend Tics for task -= EXECUTIC
- ELSE Task Suspend Tics for task = 0 ENDIF ENDFOR
- //Find the highest priority task with zero suspend interval.
- FOR Task=0 TO Main Task IF Task Suspend Tics for task == 0 Current Task = Task break ENDIF ENDFOR
- // The Current Task Pointer points to the new (or current) task.
- Current Task Pointer = Pointer to TCB of Current Task EXECUTIC = 0; ENDIF
- @

Специфікація процесу поновлення контексту програмної задачі

35

- @IN = Current Task Pointer @IN = Task Stack Pointer
@IN = Task Context
- @OUT = Processor Context
- @PSPEC Restore Task Context
- The microprocessor is in IRQ mode.
- Restore the User Mode Stack Pointer (sp of the restored task) from the TCB Task Stack Pointer for restored task (pointed to by Current Task Pointer)
- Restore the User Mode Processor Context (the interrupt task Task Context)R0-r14, SPSR&pc from the stack of the restored task.
- Store the User Mode Stack Pointer (the new Task Stack Pointer of the interrupted task) to the TCB of the interrupted task (pointed to by User Mode Stack Pointer).
- Switch the processor to User Mode.
- @

Припинення задач

36

- Задача може тимчасово зупинити себе в любую мить, але звичайно зупинка виконується коли завершено виконання всіх операцій для поточної її активації. Після зупинки задача з найвищим пріоритетом отримує управління мікропроцесору.

Схема припинення задачі (Parent: GPS Architect OS)

37

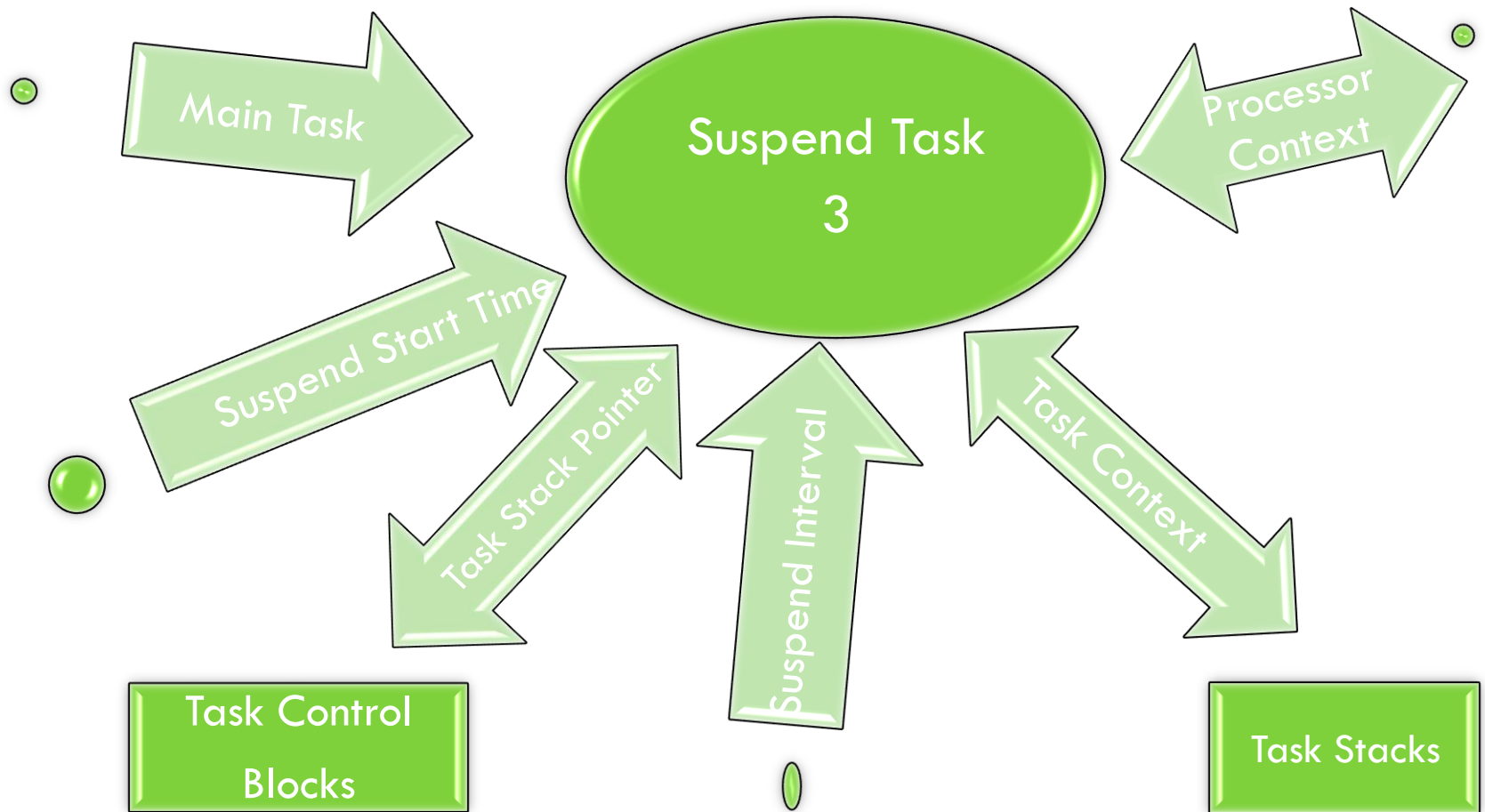
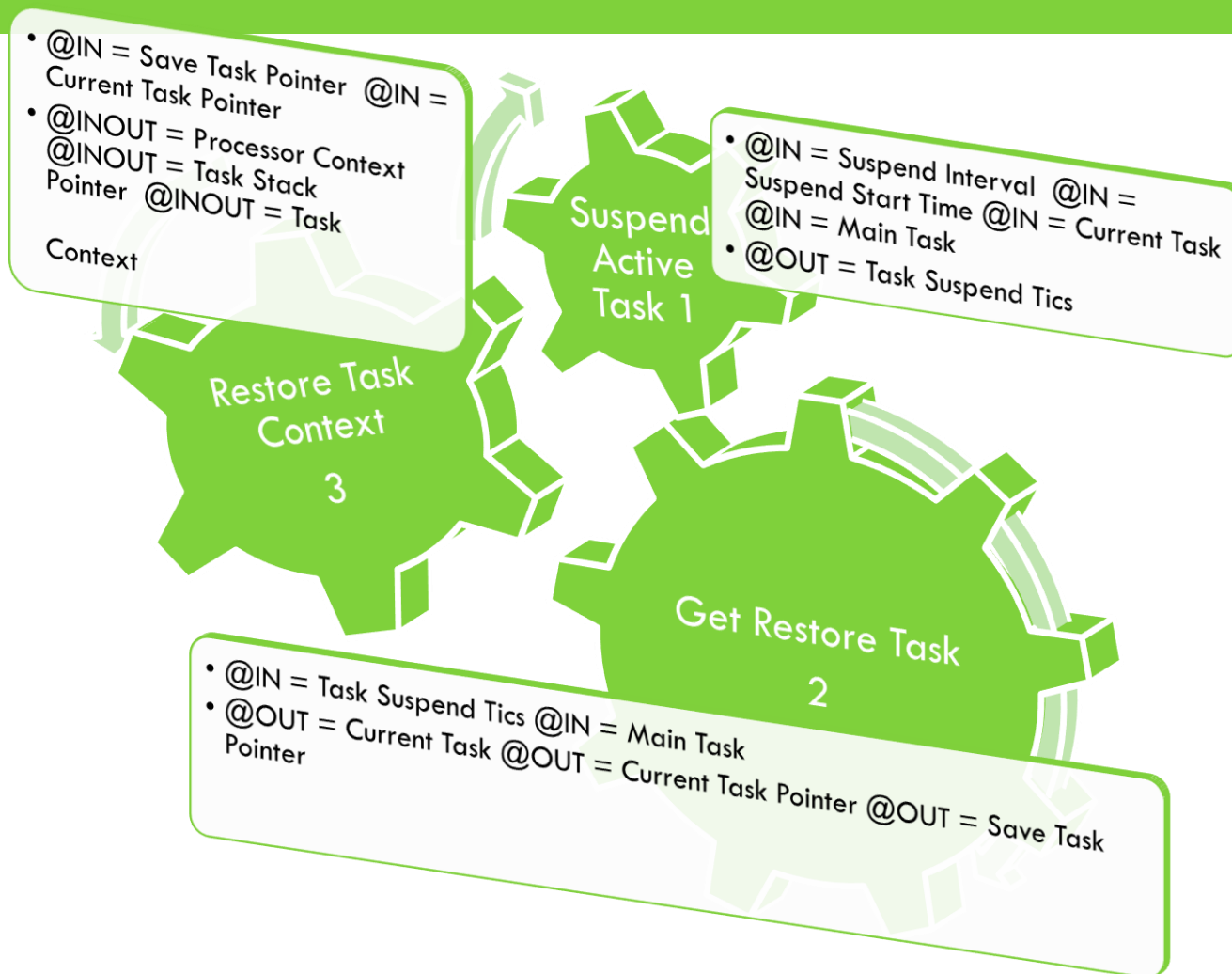


Схема припинення активної задачі

38



Припинення активної задачі

39

- @IN = Suspend Interval @IN = Suspend Start Time
 @IN = Current Task @IN = Main Task
- @OUT = Task Suspend Tics
- @PSPEC Suspend Active Task
- //Can't suspend MAIN
- IF Current Task == Main Task
- RETURN
- Get current time.
- IF (current time – Suspend Start Time) < (Suspend Interval + 1)
- Task Suspend Tics for Current Task =
- Suspend Interval - (current time -Suspend Start Time)
- ELSE
- Task Suspend Tics for Current Task = 1
- ENDIF
- @

Специфікація процесу визначення активної задачі

40

- @IN = Task Suspend Tics @IN = Main Task
- @OUT = Current Task @OUT = Current Task Pointer
- @OUT = Save Task Pointer
- @PSPEC Get Restore Task
- Save Task Pointer = TCB Pointer of Current Task
- FOR Task = 0 TO MainTask)
- IF Task Suspend Tics for Task ==0)
- Current Task = Task
- break;
- ENDIF
- ENDFOR
- Current Task Pointer = TCB Pointer of Current Task
- @

Специфікація процесу збереження та поновлення програмної задачі

41

- @IN = Save Task Pointer @IN = Current Task Pointer
- @INOUT = Processor Context
@INOUT = Task Stack Pointer @INOUT = Task Context
- @PSPEC Save And Restore Tasks
- Store the Processor Context (Task Context of the save task):
 - R0-r14, SPSR&pc
 - to the stack pointer of the TCB Save Task Pointer.
 - Store the new stack pointer to the TCB Task Stack Pointer of the save task.
- Restore the processor context(Task Context of the restore task):
 - R0-r14, SPSR&pc
 - from the stack of the TCB pointed at by Current Task Pointer.
- @