Методология за синтез на паралелни алгоритми

Особености на алгоритмичния синтез

- Алгоритмите, предназначени за паралелна обработка, се различават съществено от алгоритмите, предназначени за последователна обработка. За да може да се оцени и измери реалната им производителност, се налага *пренаписване на кода*.
- ***Производителността*** на паралелните алгоритми в много случаи *зависи от размера на проблема*. *Ако избраният размер на проблема не е оптимален, оценката на производителността не е коректна*.

- *За подобни паралелни архитектури производителността може да варира в широки граници*. При това топологията и динамичните характеристики на вътрешносистемната комуникационна мрежа представлява критичен фактор, определящ производителността.
- Освен необходимостта от създаването на нови техники за синтеза на алгоритмите, паралелната обработка изисква *да се преосмислят нашите схващания за компютърните системи, езиците за паралелно програмиране и методите за изчисления*.

- ***Парадигмите за синтез на паралелни алгоритми дефинират начините за структуриране на паралелните алгоритми за изпълнението им от паралелна компютърна система.***
- В рамките на една и съща паралелна програма *няколко парадигми* могат да бъдат *комбинирани* по различни начини.
- Паралелната компютърна система трябва да бъде достатъчно ***гъвкава***, за да поддържа разнообразните алгоритмични парадигми.

Съществуват три основни подхода при синтеза на паралелни алгоритми:

1. ***Паралелизация*** на съществуващите последователни алгоритми или модифицирането на тези техни части, в които има заложен потенциален паралелизъм.
2. ***Синтез на нов паралелен алгоритъм***, който може да се адаптира към особеностите на паралелната архитектура.

3. Синтез на нов паралелен алгоритъм от съществуващ паралелен алгоритъм.

От теоретична гледна точка алгоритмите могат да бъдат трансформирани от последователна в паралелна форма или от една паралелна форма в друга.

Алгоритмичната трансформация е валидна само ако тя поддържа алгоритмична еквивалентност.

В резултат на тази алгоритмична трансформация могат да бъдат дадени отговори на следните въпроси:

- ***Коя архитектура*** е най-подходяща за решаването на даден проблем?
- ***Каква ефективност*** при решаването на даден проблем може да се получи при използването на дадена паралелна компютърна архитектура?

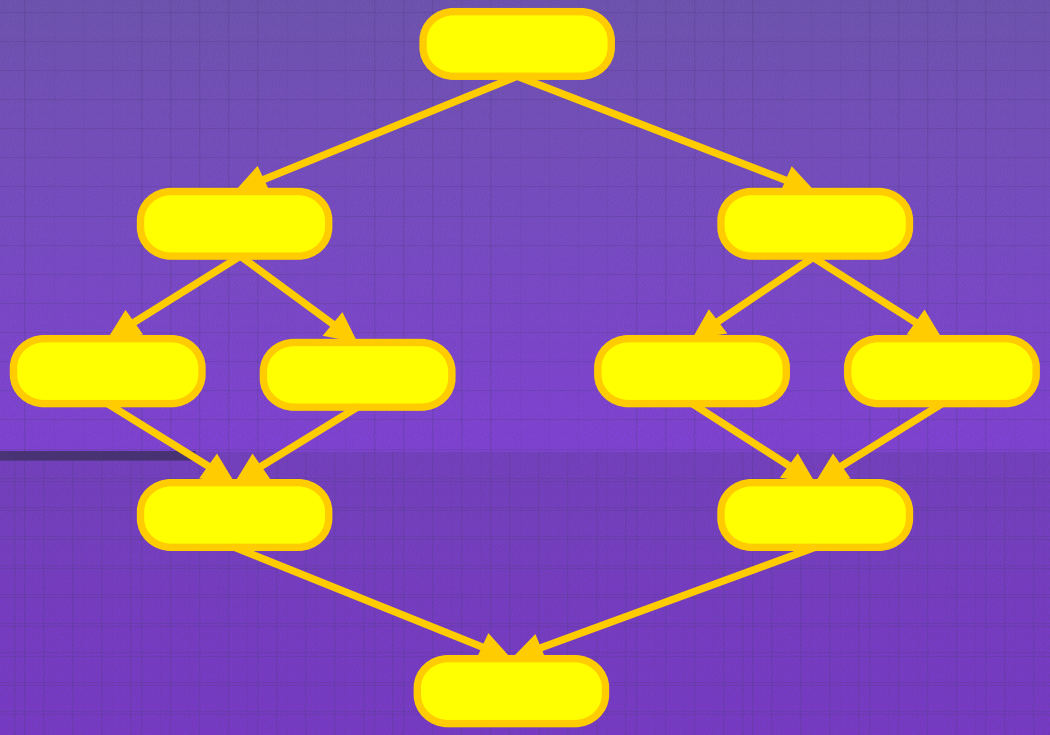
Основни парадигми за синтеза на паралелните алгоритми:

- ***"Разделяй и владей"*** - представлява основата на паралелното програмиране

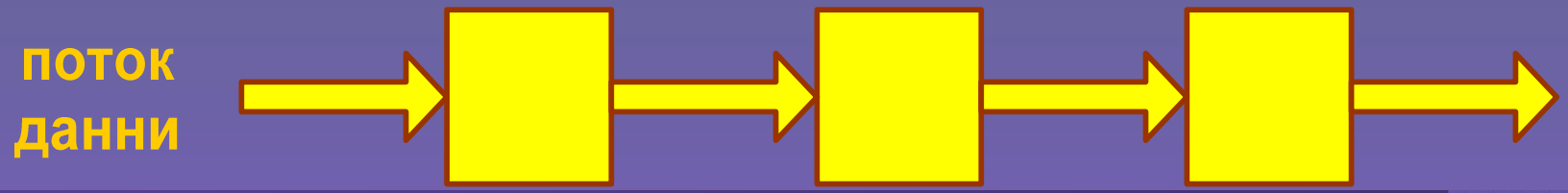
  Декомпозиция по данни

  (data partitioning) или (domain decomposition)

  Функционална декомпозиция

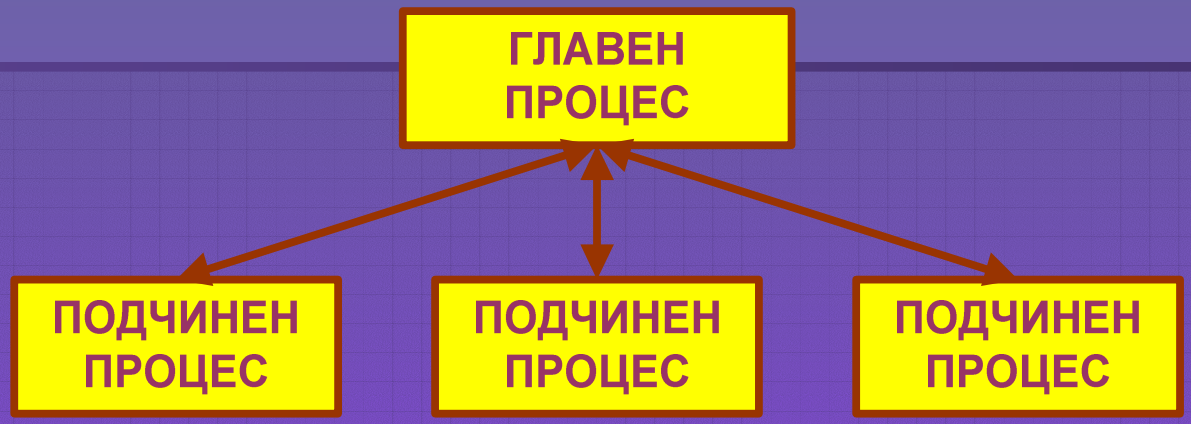


•Конвейеризация на паралелните изчисления



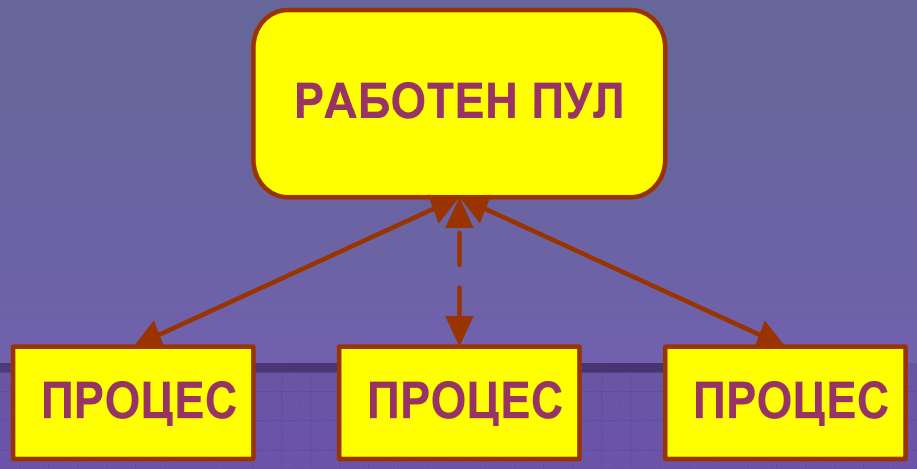
- ✓ множество процеси формират *виртуален конвейер*
- ✓ припокриване на едновременната обработка на различните фази на множество процеси
- ✓ най-често се използва *при MIMD алгоритмите*, но също така и при някои SIMD алгоритми

- ***Главен-подчинен*** (master-slave), известна още като “processor farms”



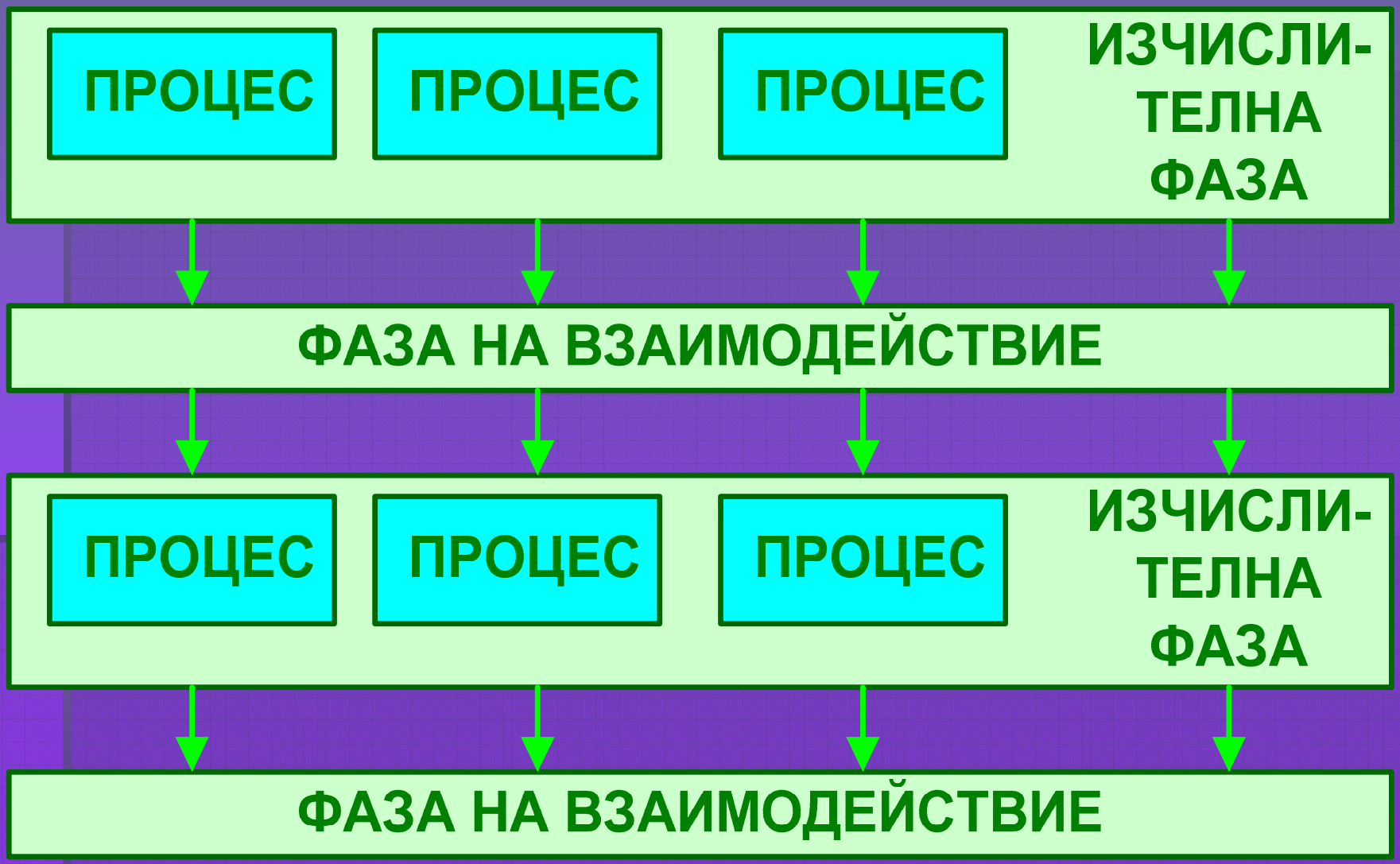
- ✓ *координацията се осъществява от главния процес (тясно място)*
- ✓ главният процес изпълнява основната последователна част на програмата
- ✓ разпростира множество подчинени процеси, обработващи паралелния работен товар

•Работен пул



- *Пулът се реализира като глобална структура от данни*
- *Работният пул може да бъде ненаредено множество, опашка или приоритетна опашка*
- **Всеки свободен процес може да вземе произволен работен товар от пула и да го изпълни**
- **Изпълнението на паралелната програма завършва, когато се изчерпи изчислителния товар в работния пул**
- ***Добро балансиране на товара***
- ***Недостатък* - трудната имплементация на работния пул, особено при модела с обмен на съобщения**

•Фазово-паралелна алгоритмична парадигма

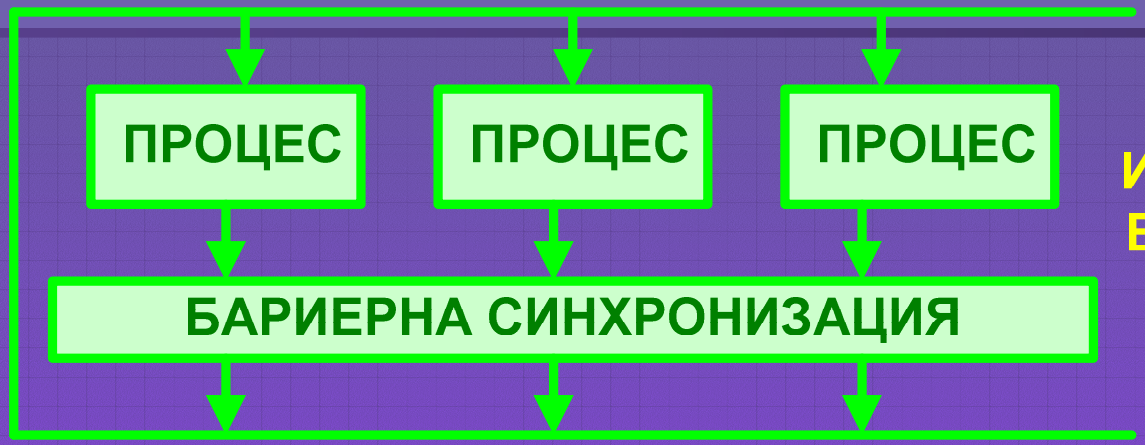


- *Слабо синхронна парадигма* или парадигма с “дневен ред” (agenda paradigm)
- Изпълнението на паралелната програма включва определен брой суперстъпки
- Всяка *суперстъпка* включва две фази: изчислителна фаза и фаза на взаимодействието (координираща фаза)
- Двете фази се изпълняват непрекъснато и алтернативно

- ✓ През координиращата фаза процесите взаимодействат чрез *синхронни операции като например бариерна синхронизация или блокираща комуникация*
- ✓ ***Недостатъци:***

- взаимодействието на процесите не се припокрива с изчисленията
- трудности при баланса на паралелния изчислителен товар

•Синхронни и асинхронни итерации



- ✓ *специален случай на слабо синхронната парадигма*
- ✓ *суперстъпките представляват последователност от итерации в рамките на цикъл*

- При *синхронни итерации* (например с използване на бариерна синхронизация), никой от паралелните процеси не може да започне (i+1)-тата итерация на цикъла преди всички процеси да са завършили i-тата итерация на цикъла.
- Парадигмата с *асинхронни итерации*, обаче, дава възможност даден процес да продължи с изпълнението на следващата итерация без да изчаква останалите процеси да завършат изпълнението на текущата итерация.

В общия случай асинхронните итерационни алгоритми са по-бързи от съответните им синхронни варианти !

Съответствия между алгоритмичното и архитектурното пространства

Паралелен алгоритъм	Паралелна архитектура
Грануляритет	Сложност
Степен на паралелизъм	Работен режим
Структури данни	Структура на паметта
Комуникационна среда	Системна комуникационна мрежа
Размер на алгоритъма	Брой процесори
Подход при програмирането	Категория на компютърната архитектура

Целта е да се дефинира такава ***обобщена архитектурна рамка***, която *да не допуска нежелателни ограничения на архитектурата при алгоритмичния синтез*.

За синтеза на оптималния алгоритъм е необходимо да се направи селекция на архитектурната конфигурация, организацията на паметта, размера на секциите, режима на работа и мрежовата конфигурация.

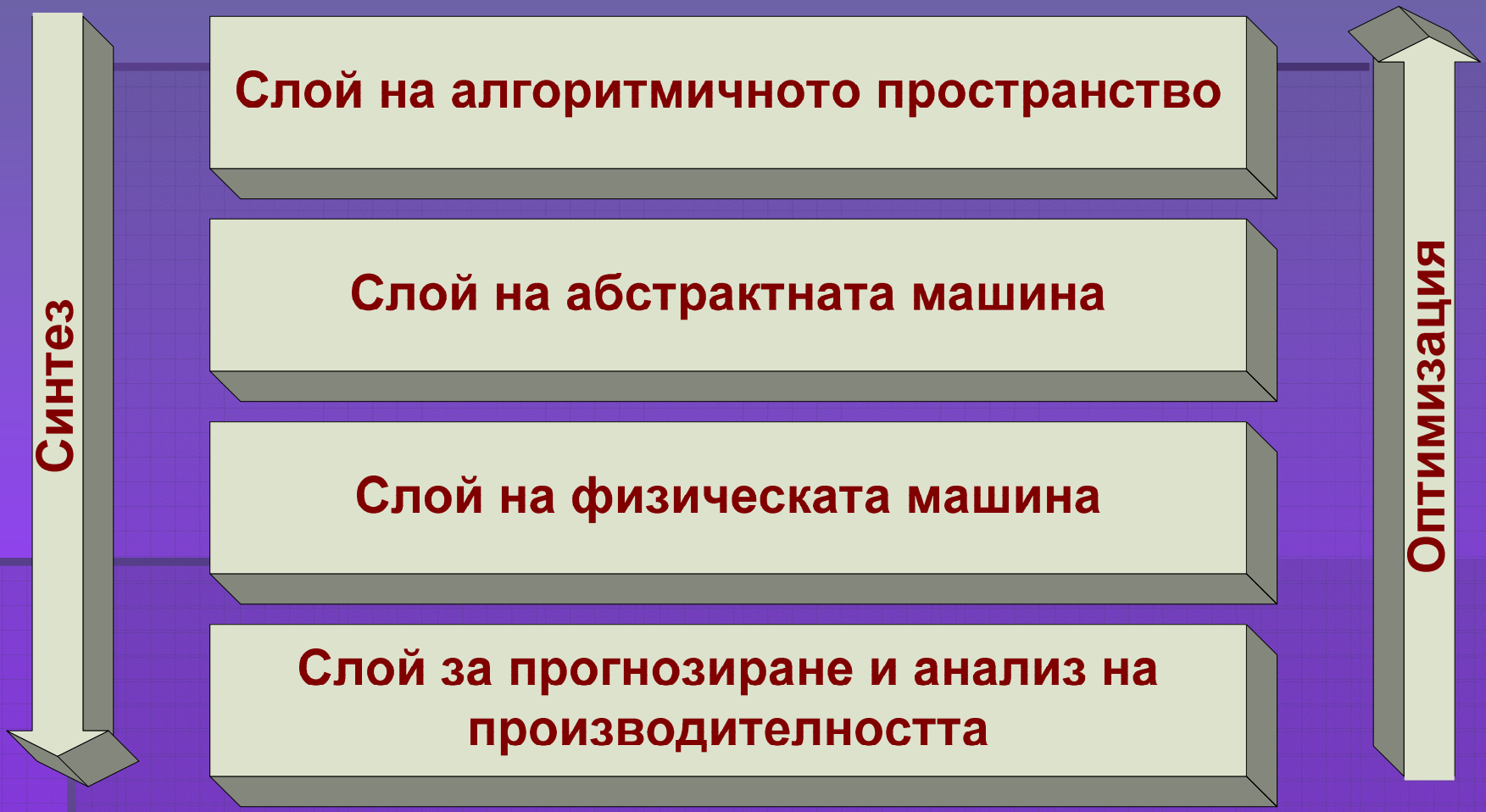
Основни аспекти на алгоритмичния синтез

- Паралелизъм
- Модулност
- Локалност
- Възможност за мащабиране
- Първо се разглеждат машинно независимите аспекти като степен на паралелизъм и възможности за декомпозиция
- След това - машинно зависимите аспекти като брой на процесорите и допълнителни разходи за комуникация и изобщо за координация на паралелните процеси

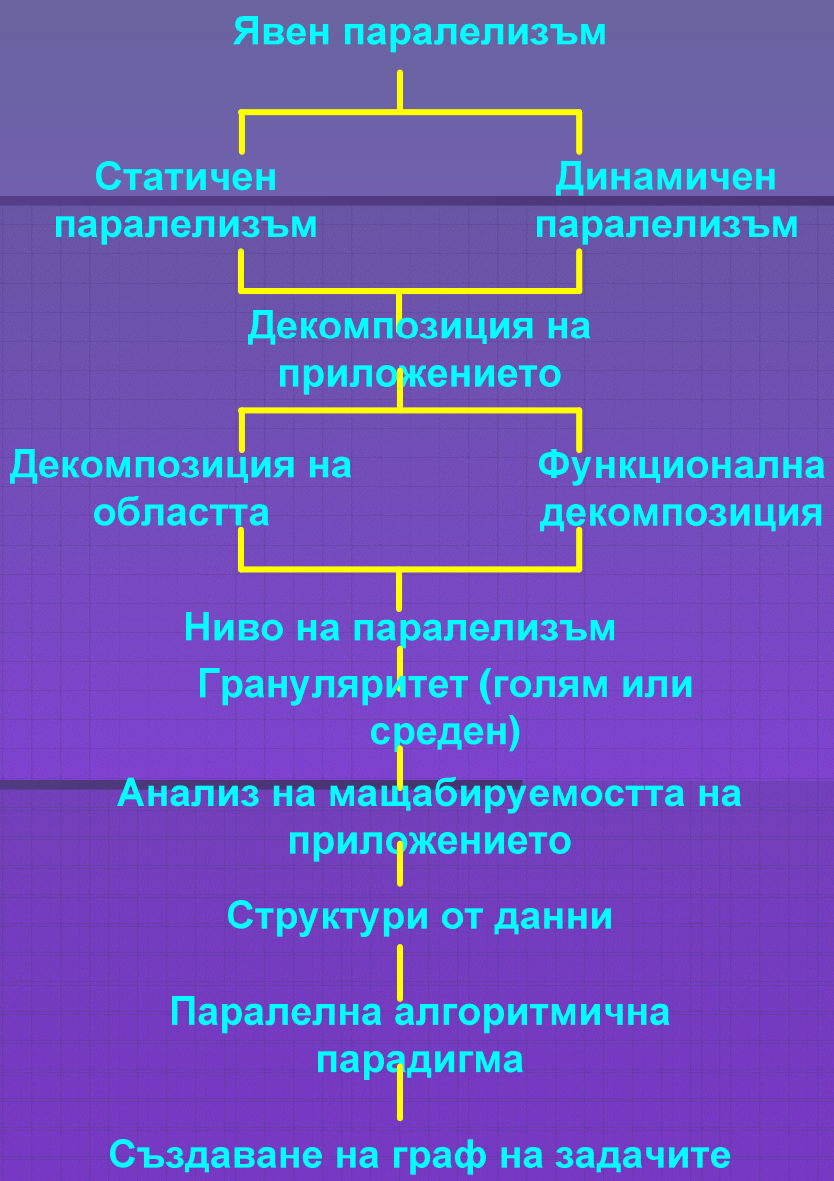
- Целта е да се изгради методология за алгоритмичен синтез като последователност от ясно определени *фази*: *декомпозиция (разделяне на задачи), комуникация, агломерация (групиране на задачите) и планиране (разпределяне на задачите)*.
- В първата и втората фаза се анализират степента на паралелизъм и възможностите за мащабиране при решаването на проблема, докато при втория етап се вземат предвид разходите за комуникация и се анализира производителността.

Архитектурни аспекти на клъстерите

Йерархична многослойна процедура за алгоритмичен синтез

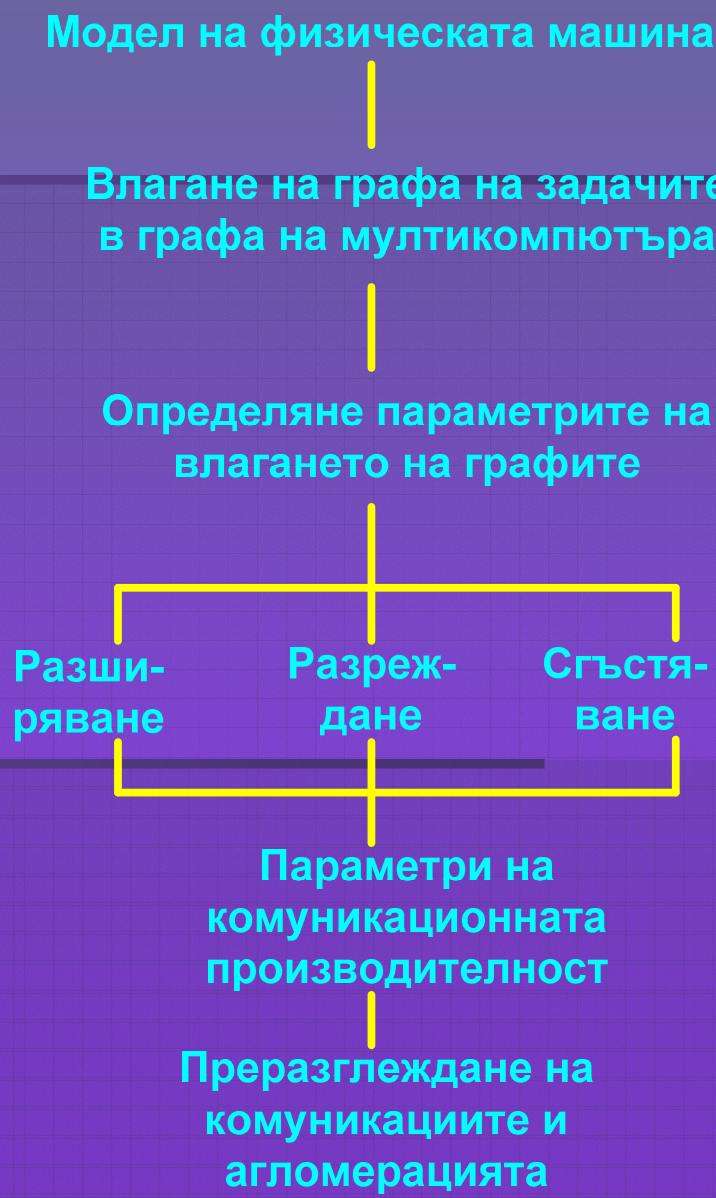


Слой на алгоритмичното пространство

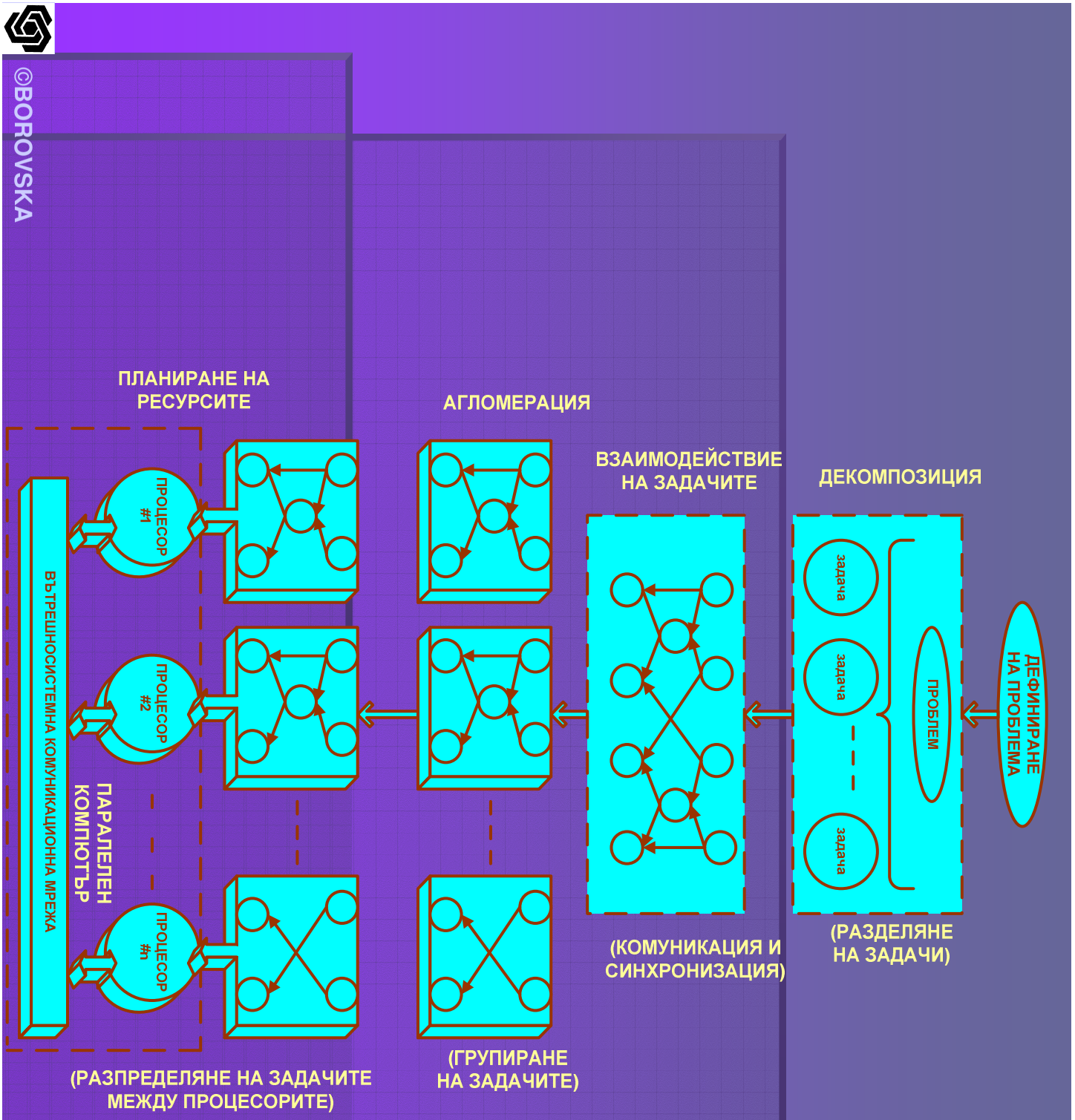


Слой на абстрактната машина

Слой на физическата машина



Слой за прогнозиране и анализ на производителността


ДЕКОМПОЗИЦИЯ
ДЕФИНИРАНЕ НА ПРОБЛЕМА
ПРОБЛЕМ
задача
задача
задача
(РАЗДЕЛЯНЕ НА ЗАДАЧИ)
ВЗАИМОДЕЙСТВИЕ НА ЗАДАЧИТЕ
(КОМУНИКАЦИЯ И СИНХРОНИЗАЦИЯ)
АГЛОМЕРАЦИЯ
(ГРУПИРАНЕ НА ЗАДАЧИТЕ)
ПЛАНИРАНЕ НА РЕСУРСИТЕ
(РАЗПРЕДЕЛЯНЕ НА ЗАДАЧИТЕ МЕЖДУ ПРОЦЕСОРИТЕ)
ПАРАЛЕЛЕН КОМПЮТЪР
ВЪТРЕШНОСИСТЕМНА КОМУНИКАЦИОННА МРЕЖА
ПРОЦЕСОР #1
ПРОЦЕСОР #2
ПРОЦЕСОР #n

Фаза на разделяне

- *броят на процесорите в целевата паралелна компютърна архитектура се игнорира*
- дефинирането на *максимален брой малки паралелни* задачи с цел да се постигне декомпозиция на фини гранули
- По принцип, добрият метод на декомпозиция осигурява *ефективно разделяне както на кода, така и на съответните данни, на малки гранули*

методи за декомпозиция на проблема

1. *Декомпозиция на данните*, свързани с проблема, при която разделянето първо се осъществява по отношение на данните и след това се определят съответните изчисления, свързани с отделните дялове от данните.
2. *Функционална декомпозиция*, при която изчисленията се разделят на отделни задачи и впоследствие се разделят данните.

фаза на взаимодействие на задачите

- ***Цели:***

✓ да се изгради адекватната комуникационна структура на приложението

✓ да се конструират съответните алгоритми за координация на изпълнението на задачите

✓ Анализ на възможността за възникване на *хазарти*

- Важна роля играят структурата на комуникационните канали и механизма за обмен на съобщения
- При структури с обмен на съобщения се формират съобщенията, които трябва да се изпратят през и получат от каналите
- Операциите за комуникация се организират по такъв начин, че да се осигури *едновременното изпълнение на максимален брой изчислителни операции и операции за комуникация*

Фаза на агломерация

- *преход от абстрактния към конкретния синтез*
- задачите се групират като се вземат предвид особеностите на имплементацията и факторите, определящи производителността
- ревизиране на получените резултати в първите две фази с оглед ***максимално адаптиране на структурата на алгоритъма към особеностите на целевата компютърна архитектура***

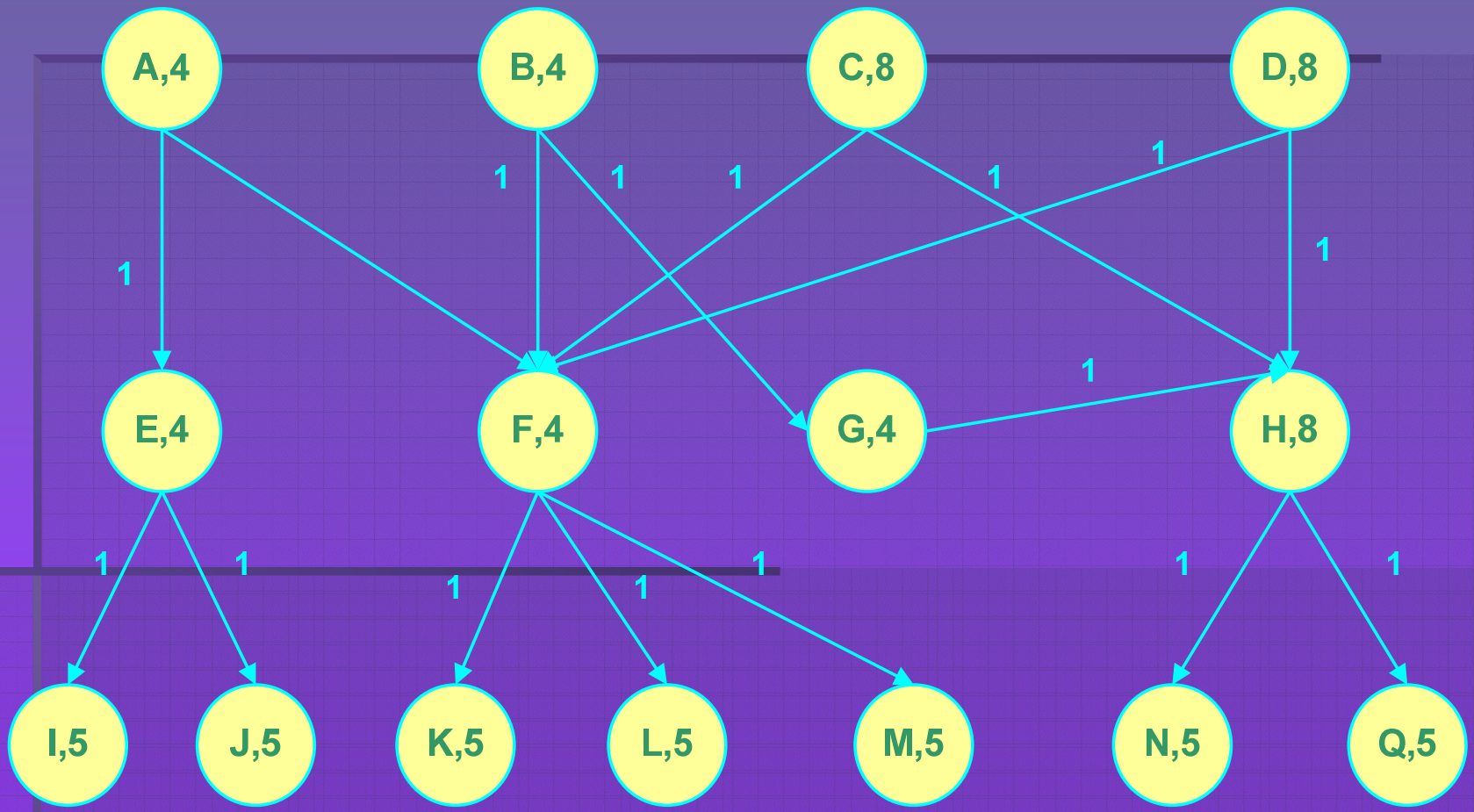
фаза за планиране

- ***Цел***: такова разпределение на задачите между процесорите, което осигурява най-висока производителност
- максимално използване на изчислителните ресурси на паралелния компютър, по възможност равномерно натоварване на процесорите и минимизиране на разходите за комуникация
- ***Планирането*** може да се осъществява *статично* (преди старта на изчислителния процес) или *динамично* (в хода на изпълнението на паралелната програма)

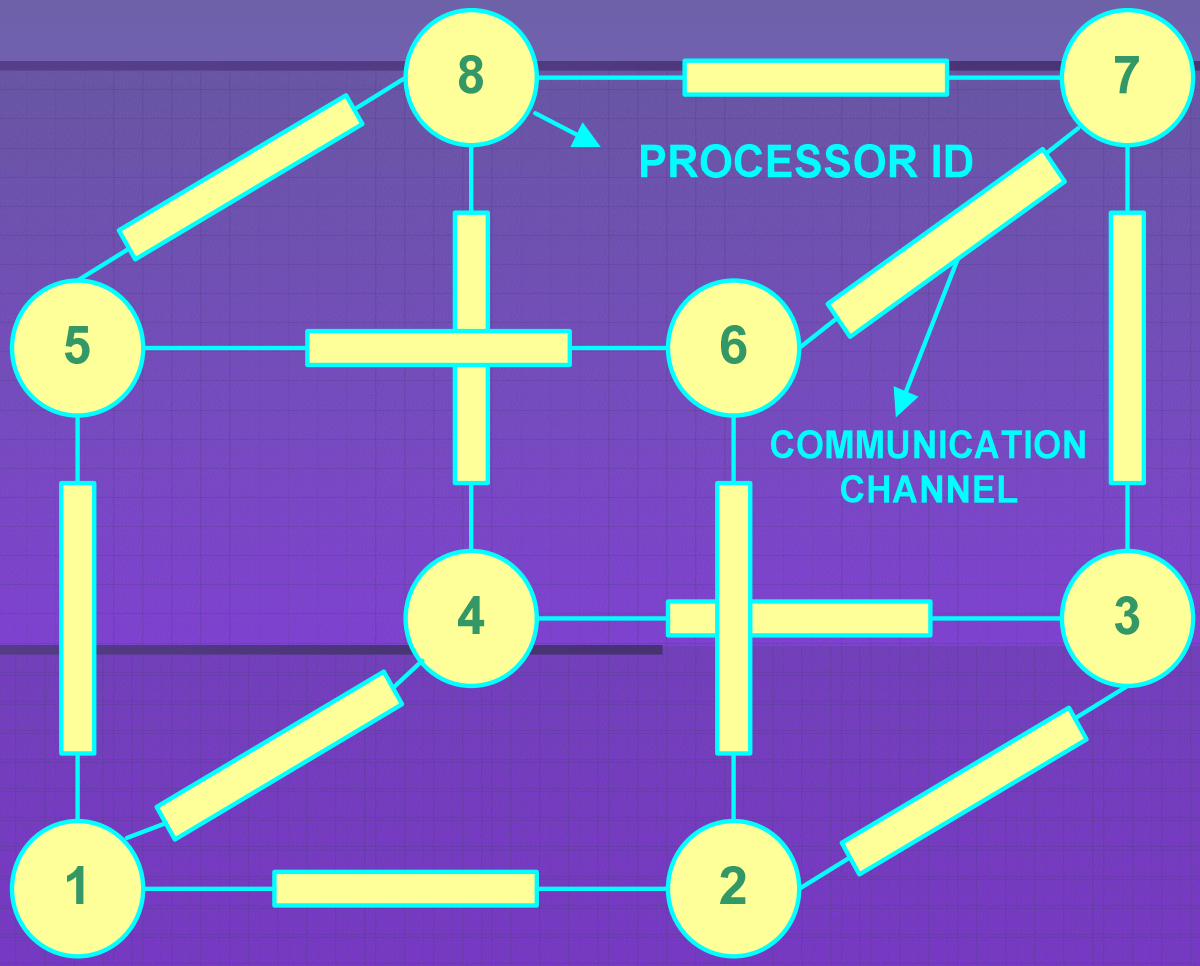
- *проблемът за планиране при паралелните алгоритми е от сложност NP*
- в тази фаза *основните цели са повишаване степента на паралелизъм и локалността*
- да се конструира графа на зависимостите при изчисленията т.е. ***граф на задачите:***

насочен граф, възлите на който представят блокове от независими операции, а дъгите представят ситуациите, при които изпълнението на даден блок операции зависи от изпълнението на други блокове от операции.

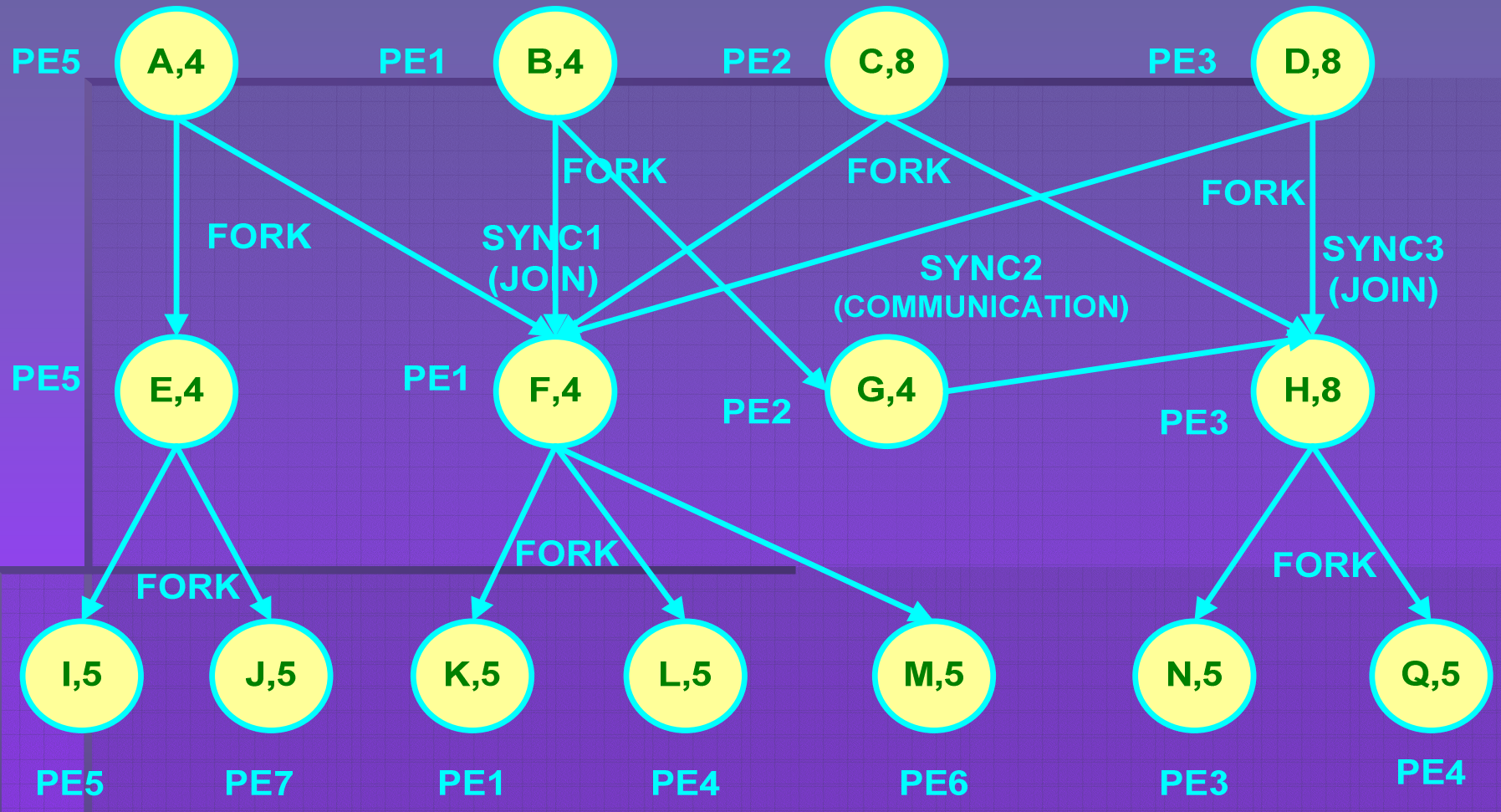
Граф на задачите



Граф на компютъра



Планиране на ресурсите



Достойнствата на синтезирания паралелен алгоритъм се оценяват като се анализират следните му атрибути:

- Паралелизъм на ниво данни (SIMD)
- Функционален паралелизъм (MIMD)
- Гранулярност на процесите
- Гранулярност на данните
- Степен на паралелизъм
- Еднородност на операциите
- Синхронизация
- Зависимости по данни
- Генериране и унищожаване на процесите

КРАЙ