

Конвейерно изпълнение на инструкцията



Автор: гл. ас. д-р инж. Любомир Богданов



Европейски съюз

ПРОЕКТ BG051PO001--4.3.04-0042

„Организационна и технологична инфраструктура за учене през целия живот и развитие на компетенции”

Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз

Инвестира във вашето бъдеще!



Европейски социален фонд

Съдържание

1. Изпълнение на инструкцията
2. Конвейери
3. Блокиране на конвейерите
4. Конвейери и програмен брояч
5. Предсказание на преходите (branch prediction)
6. Спекулативно изпълнение (speculation)

Изпълнение на инструкцията

Вътрешната структура на един μ PU може да се раздели на две части [1]:

- ***фронтенд** (front end) – състои се от контролен модул (включващ програмен брояч PC, регистър на инструкцията IR, регистър на състоянието STAT, и др.) и входно-изходен I/O модул (отговорен за работа с паметта);

- ***бекенд** (back end) – състои се от АЛУ и регистри с общо предназначение GPR.

Изпълнение на инструкцията

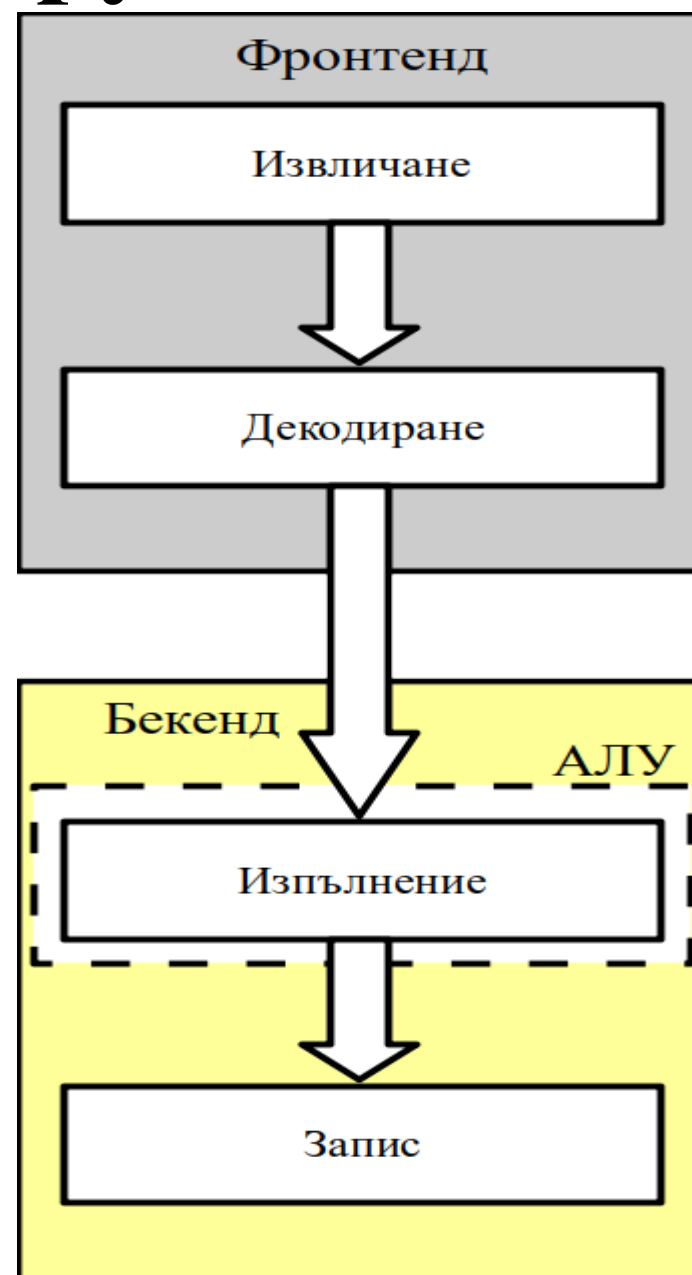
Изпълнението на една инструкция минава през /поне/ 4 фази:

- *извличане (**I**nstruction **F**etch, IF)
- *декодиране (**I**nstruction **D**ecode, ID)
- *изпълнение (**E**xecute, EX)
- *запис на резултат (**W**rite, WR)

Изпълнение на инструкцията

Извличането и декодирането се извършват от фронтенда.

Изпълнението и записът на резултата се извършват от бекенда.



Изпълнение на инструкцията

Допълнително записът на резултата може да се раздели на две фази:

***вътрешен запис (Write Back, WB)** – след завършване изпълнението на инструкцията, резултатът се записва обратно в регистровия файл на ядрото (или още - работните регистри, GPR);

***външен запис (Memory Access, MEM)** - след завършване изпълнението на инструкцията, резултатът се записва във данновата (в повечето случаи – RAM) памет.

Изпълнение на инструкцията

Едно типично изпълнение на инструкцията е показано на следващия слайд.

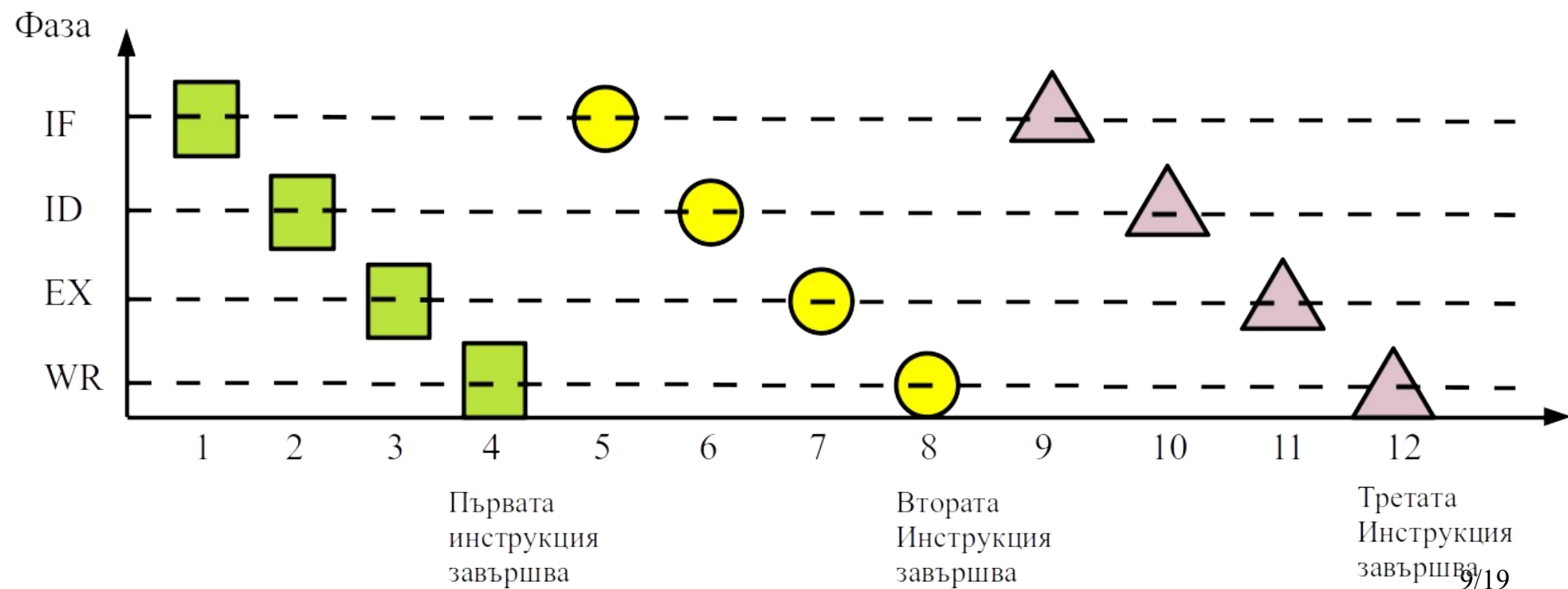
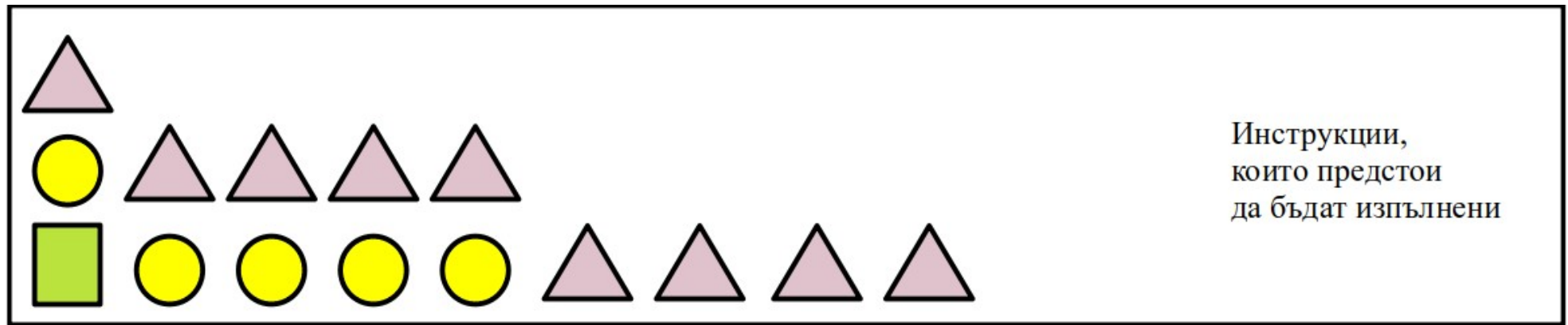
Забележете, че по време на всеки такт само е активна само една от фазите.

Такъв вид изпълнение води до много ниско работно натоварване на отделните модули на μ PU, докато една инструкция мине през всичките 4 етапа.

В примерът на следващия слайд инструкцията ще бъде изпълнена за 12 такта.

За увеличение на производителността на един μ PU, при запазване на тактовата честота, се реализират т.нар. **конвейерни микропроцесори** (pipelined microprocessors).

Изпълнение на инструкцията



Не-конвейерно изпълнение на инструкция

Конвейери

Конвейерно изпълнение на инструкция – отделните микропроцесорни модули, отговорни за IF, ID, EX и WR са автономни и **не** трябва да изчакват всяка инструкция да завърши изпълнението си преди да се захване следващата.

Чрез увеличаване сложността на хардуера и реализирането на конвейери може да се постигне по-висока производителност без да се увеличава работната честота на ядрото.

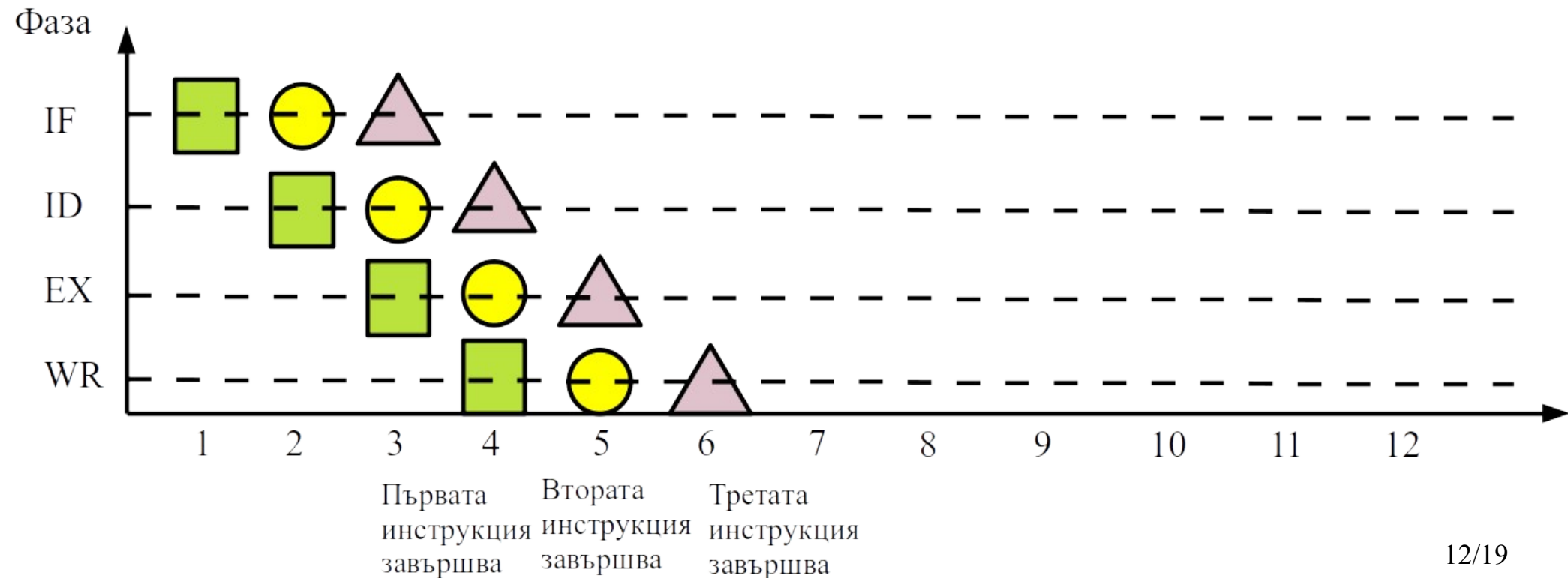
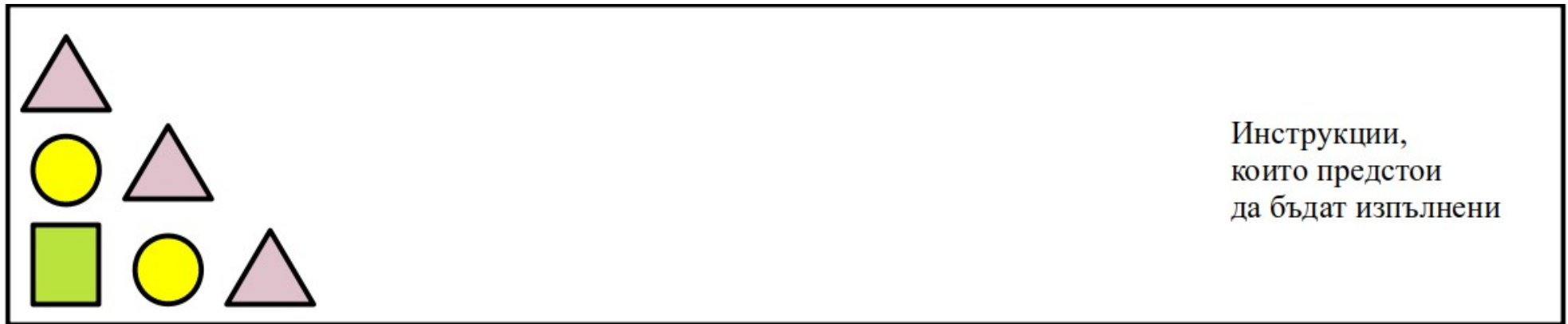
Конвейери

На следващият слайд е показано изпълнението на инструкции на конвейер.

- *Процесът започва с извличане (IF1) на първата инструкция.
- *След това се преминава във фаза декодиране (ID1). В същия времеви слот втора инструкция започва да се извлича (IF2).
- *В третата фаза се изпълнява първата инструкция (EX1). В същия времеви слот втората инструкция се декодира (ID2), а трета бива извличана (IF3).

Трите инструкции се изпълняват за 6 такта или два пъти по-бързо от варианта с не-конвейерния μ PU.

Конвейери



Конвейерно изпълнение на инструкция

Блокиране на конвейерите

Някои от инструкциите отнемат повече от един такт при преминаването си в дадена фаза. Това води до т.нар. **блокиране на конвейера** (или още “задавяне” на конвейера, pipeline stall).

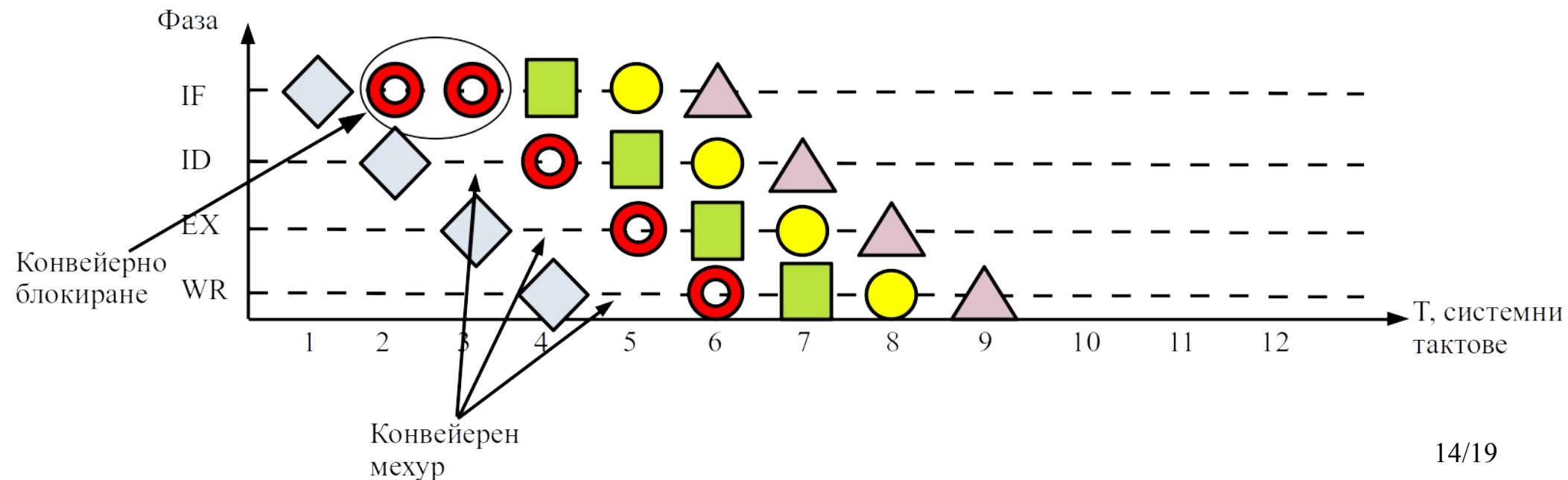
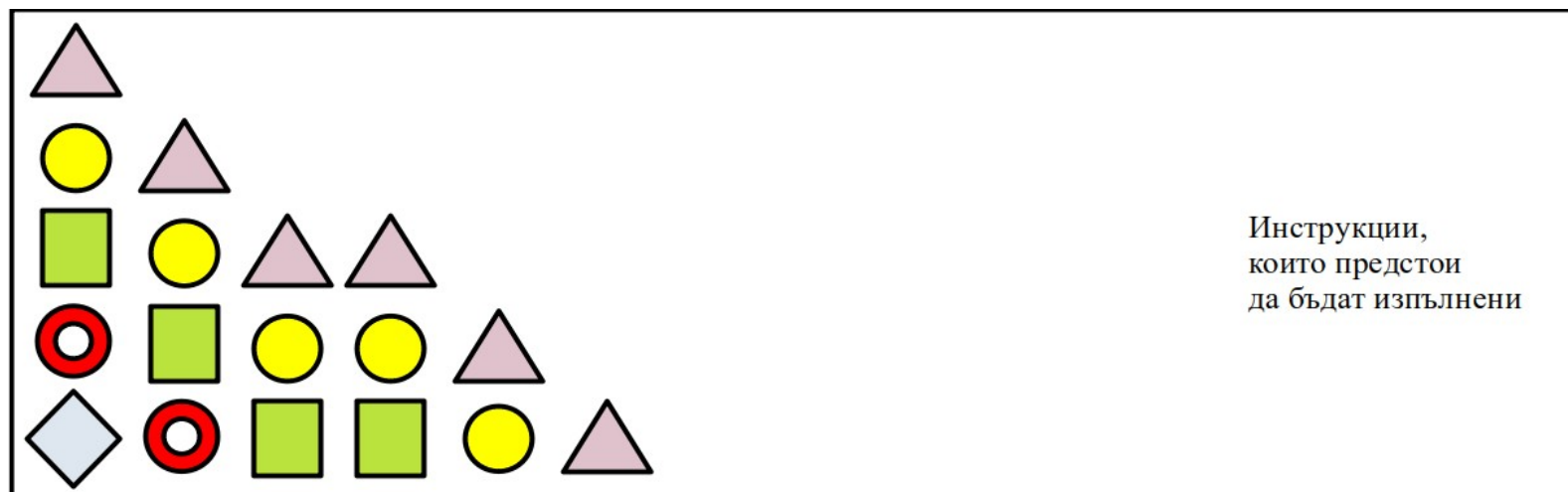
*Когато една инструкция се забави, забавят се и всички инструкции след нея.

*Когато една инструкция се забави, инструкциите преди нея продължават в следващите фази, докато не завършат изпълнението си.

Това води до т.нар. **мехури в конвейера** (pipeline bubble), които представляват неизползвани модули на μ PU.

Следващият слайд илюстрира вариант на блокиране, когато една от инструкциите заеме два такта във фазата на извличане.

Блокиране на конвейерите



Блокиране на конвейерите

На по-следващите два слайда са показани графики от [1]. Те изобразяват производителността на μ PU при 2-тактово и 10-тактово блокиране на конвейера.

Производителността на един μ PU може да се измери с параметрите:

- *тактове-за-инструкция (**Cycles Per Instruction, CPI**)

- *инструкции-за-такт (**Instructions Per Cycle, IPC**)

Тези параметри показват средния брой микропроцесорни тактове, необходими за изпълнението на една инструкция от дадена програма:

Блокиране на конвейерите

$$CPI = \frac{\sum_{i=1}^m n_i \cdot T_i}{N}$$

където CPI – параметър “тактове за инструкция”, m – брой видове инструкции, n_i – брой инструкции от вида i в програмата, T_i – брой тактове необходими за изпълнението на инструкция от вида i , N – брой инструкции в цялата програма, без значение от вида им.

Блокиране на конвейерите

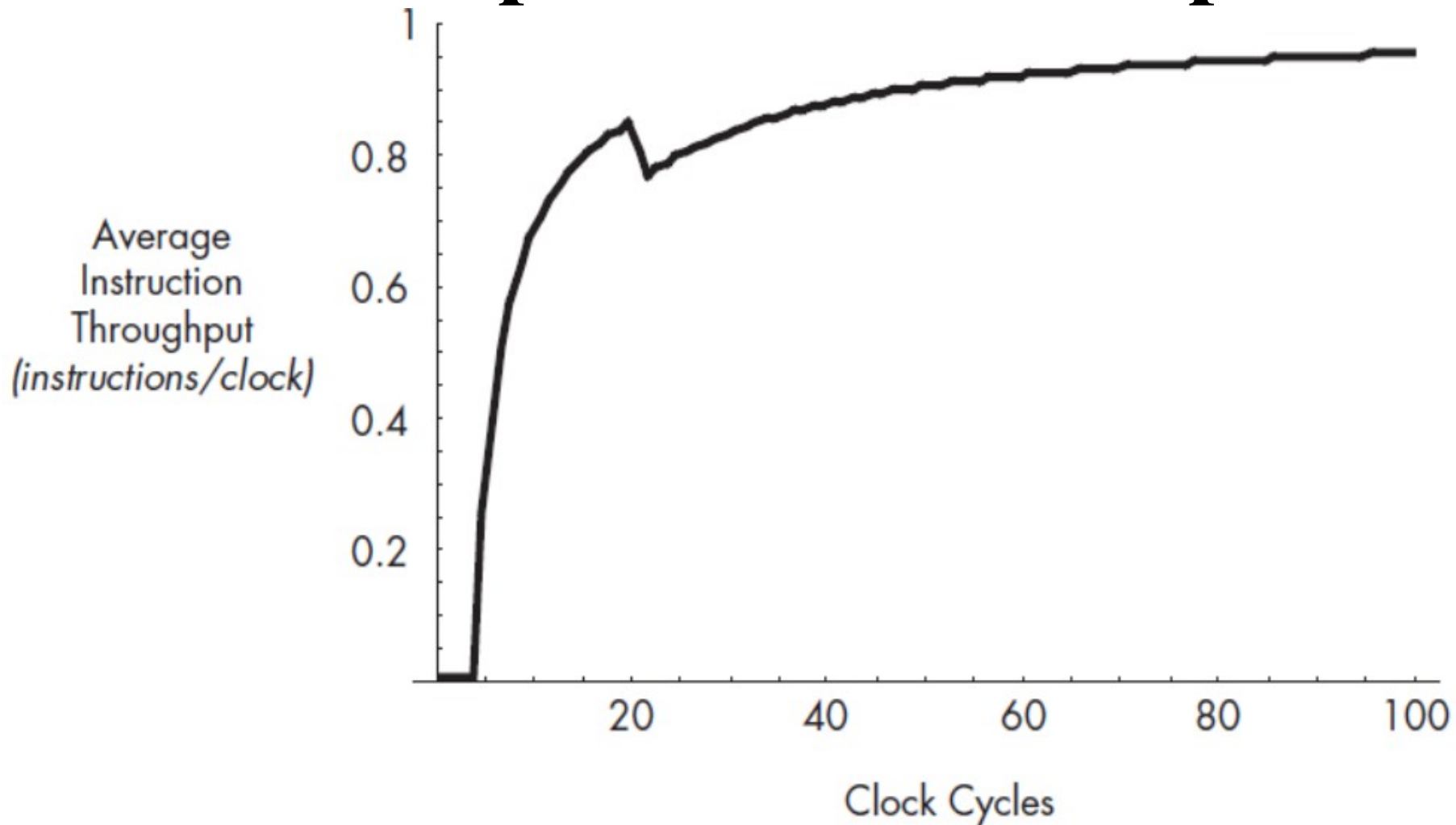


Figure 3-11: Average instruction throughput of a four-stage pipeline with a two-cycle stall

Блокиране на конвейерите

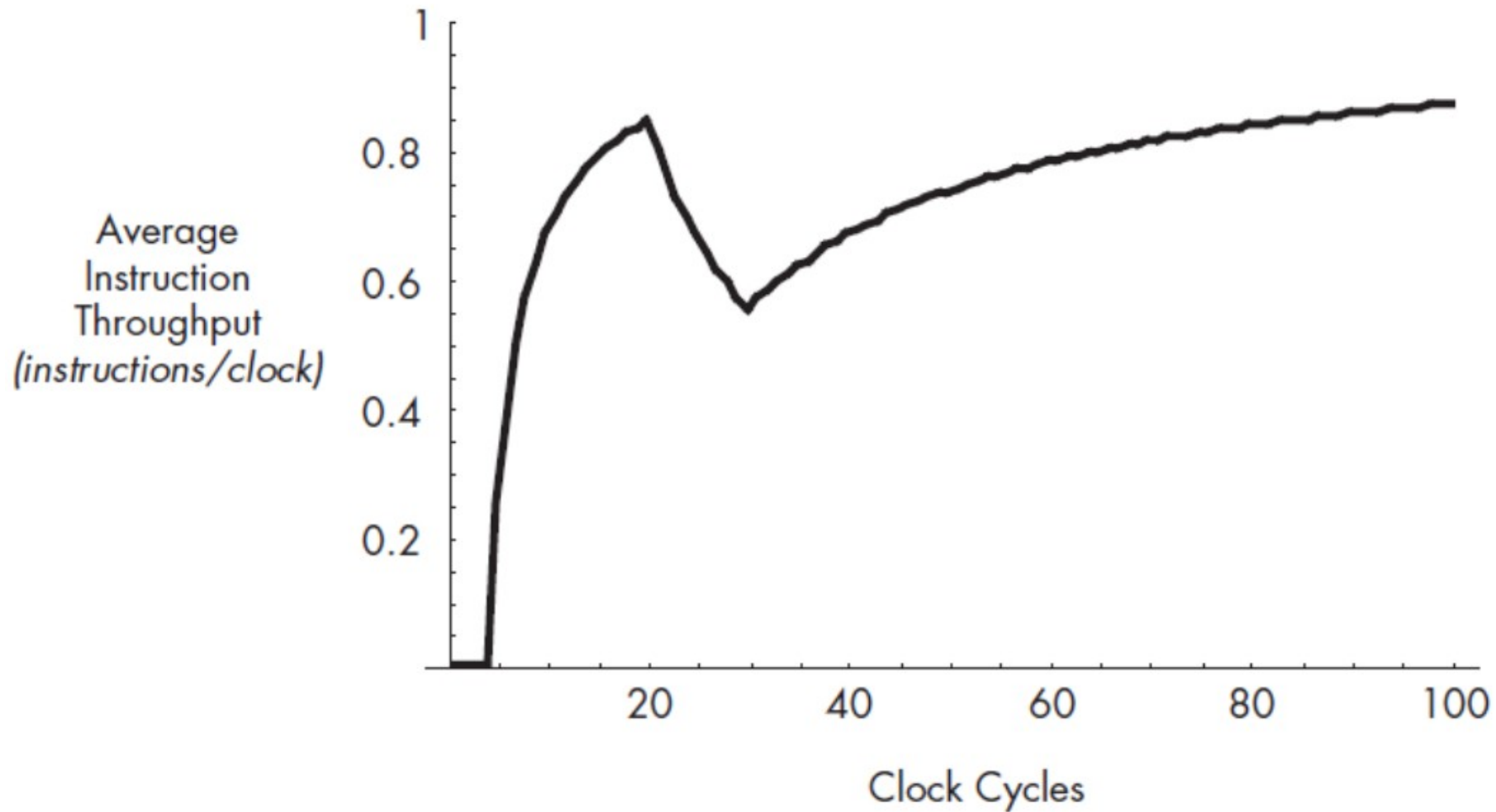


Figure 3-12: Average instruction throughput of a four-stage pipeline with a 10-cycle stall

Литература

- [1]J. Stokes, “Inside the Machine: An Illustrated Introduction to Microprocessors and Computer Architecture”, Ars Technica Library, 2007.
- [2]J. Yiu, “The Definitive Guide to ARM Cortex-M3”, Elsevier, 2007.