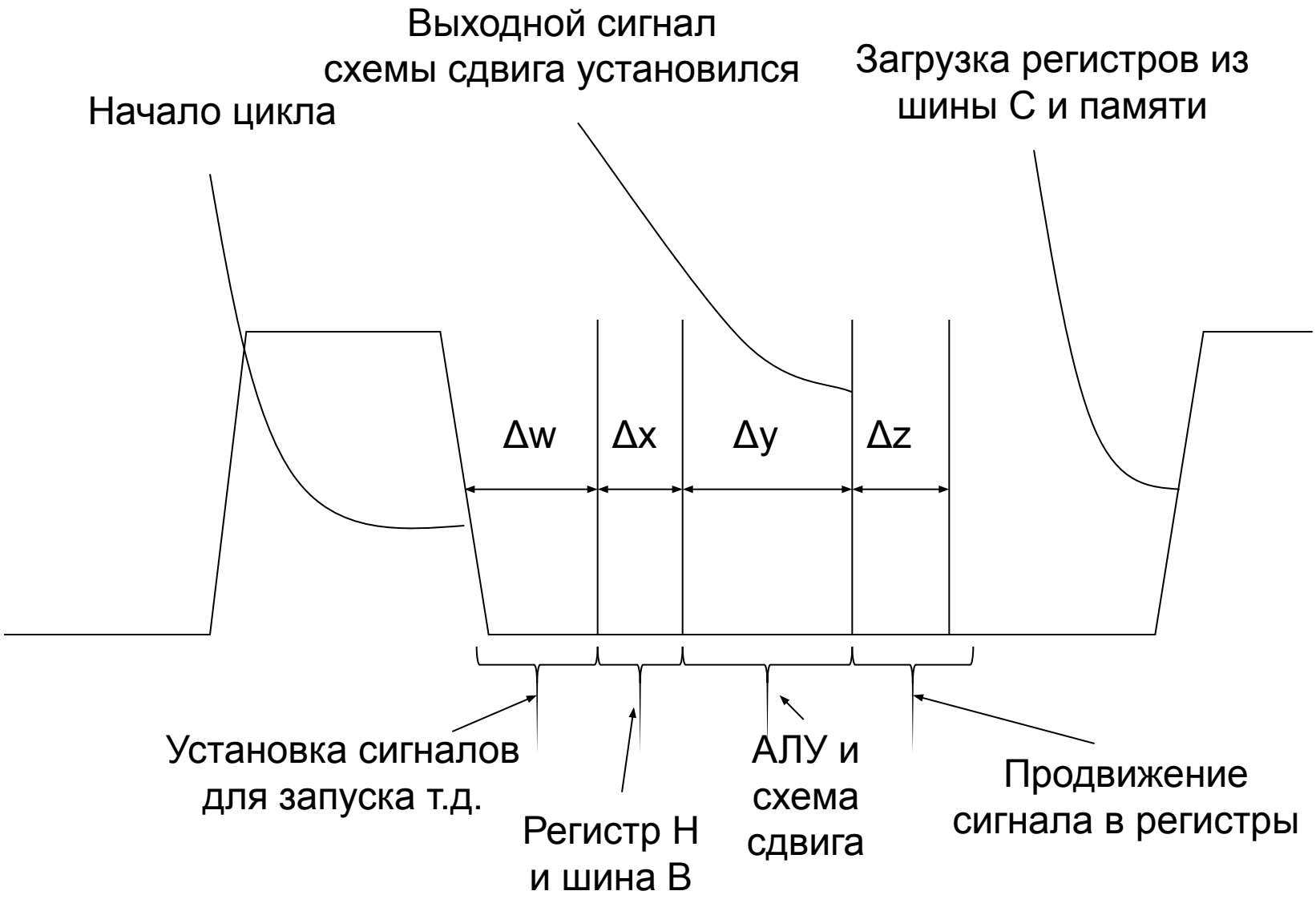
Синхронизация тракта данных



Управление трактом данных

Мы знаем что каждый цикл тракта данных можно охарактеризовать микрокомандой. Возникает вопрос: кто решает какую микрокоманду выполнить следующей?

Для этой цели наша схема тракта данных может быть дополнена так называемой управляющей памятью, которая будет содержать набор микрокоманд из которых образуются микропрограммы, соответствующие каждой команде уровня архитектуры команд. Сама по себе управляющая память – это ПЗУ.

В общем случае теперь можно сказать, что каждая команда уровня архитектуры команд на самом деле есть функция написанная на языке микрокоманд.

Выбор следующей микрокоманды

Каждая микрокоманда будет дополнена 9 адресными битами (NA) и 3 управляющими битами (JAM).

Управляющие биты будут показывать:

- надо ли использовать выходы АЛУ N и Z для образования адреса следующей микрокоманды
- надо ли использовать MBR для образования адреса следующей микрокоманды

Таким образом адрес следующей микрокоманды в управляющей памяти образуется следующим образом:

F = (NA | JAMZ) | (NA | ZAMN) | NA)

Следующий адрес = JAMPC ? F | MBR : F

Стек

Во всех языках программирования есть понятие процедур с локальными переменными. Вопрос: где они должны храниться?

Переменной нельзя предоставит абсолютный адрес в памяти. Чтобы понять почему – представьте себе рекурсивную функцию.

Поэтому для переменных резервируется особая область памяти, называемая стеком.

Организация стека

Переменные в стеке не получают абсолютных адресов! Вместо этого есть пара регистров LV и SP. LV указывает на базовый адрес (начало) локальных переменных данной процедуры. А SP указывает на их старшее слово (конец).

Структура данных между LV и SP (включая слова на которые они указывают) называется фреймом локальных переменных.

Пример:

int a(){ int a1, a2, a3; … ; b(); …d(); … }

int b(){ int b1, b2, b3, b4; …; c(); … }

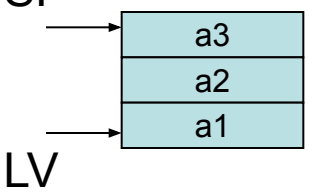
int c(){ int c1, c2; … }

int d(){ int d1; … }

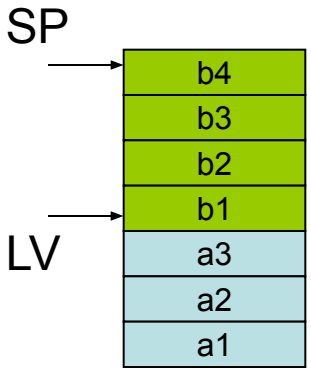
Что будет на стеке, если вызвать a()?

Содержимое стека

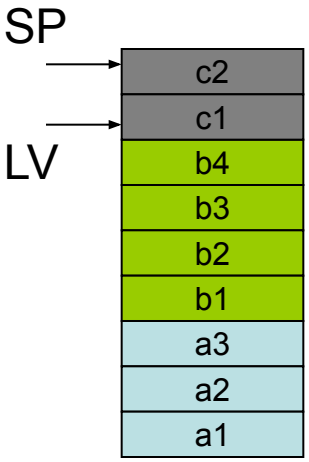
Внутри a(), но до вызова b()



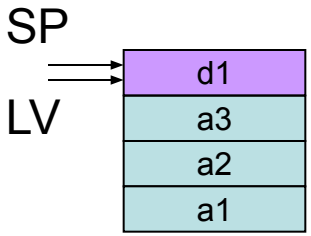
Внутри b(), но до вызова c()



Внутри c()



Внутри d()



Какие объекты в памяти могут быть связаны с каждой процедурой?

1. Набор констант
2. Фрейм локальных переменных
3. Стек операндов (для стековых машин)
4. Область процедур