

а. Джордж Бул



б. Джон Атанасов



в. Машината ABC

Фиг. 3.

През януари 1939 г. американецът от български произход Джон Атанасов (Фиг. 3б), съвместно с Клифърд Бери, създава **първия реално действащ модел** на компютър – ABC (Фиг. 3в). Приносът на Атанасов е огромен, защото вместо електро-механичните елементи на Шенън, в ABC се използва **електроника**.

## Принципи на фон Нойман

Значителна роля за оформяне на облика на съвременния компютър изиграва Джон фон Нойман. Работейки заедно с колегите си Екерт и Моучли върху машината EDVAC, Джон фон Нойман (Фиг. 4) обосновава принципите на съвременния компютър, известни като *принципи на фон Нойман*. Компютрите, построени на тези принципи, наричаме *фон Нойманови компютри*.

Накратко ще формулираме принципите, които фон Нойман обобщава и интегрира в цялостна концепция:

**Първи принцип. Компютърът е електронно устройство.** Всички операции в него се извършват от електронни схеми.

**Втори принцип. Компютърът е двоично устройство.** Данните се съхраняват в компютъра в двоичен вид. Електронните схеми също са двоични.

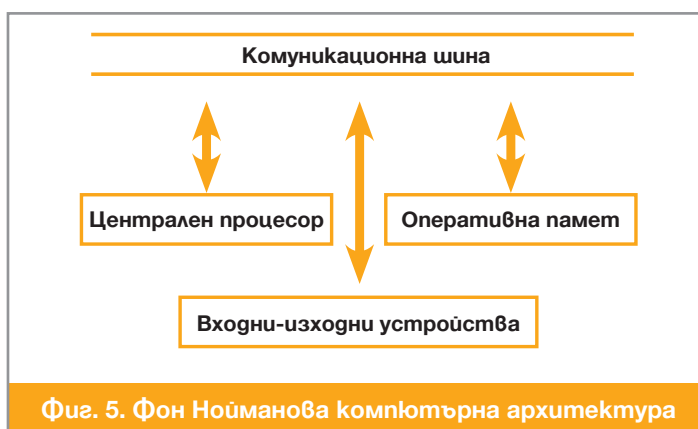
**Трети принцип. Компютърът е управляемо от програма устройство.** Програмата и входните данни се въвеждат в *компютърната памет* и се съхраняват там по време на работа. Програмите се състоят от *инструкции*, които се извличат в определен ред в *компютърния процесор*, изпълняват се над данните и резултатите се връщат обратно в паметта.

Машината EDVAC (Джон Екерт и Джон Моучли) е първата построена изцяло в съответствие с принципите на фон Нойман и от нейното название идва думата *компютър*.

Съвременните компютри са построени по една и съща принципна схема, наречена *архитектура на фон Нойман*. Тази схема (показана на Фиг. 5), както и предназначението на Централния процесор, Оперативната памет и Входно-изходните устройства познаваме от уроците по ИТ. Ще се спрем по-подробно на начина на тяхното функциониране в този и следващи уроци.



Фиг. 4. Дж. фон Нойман



Фиг. 5. Фон Нойманова компютърна архитектура

Съгласно Третия принцип на фон Нойман, компютърът е машина с програмно управление. Устройството, което осъществява програмното управление на компютъра, е Централният процесор (ЦП).

ЦП изпълнява **инструкции**, всяка от които предписва някаква **операция** с данни от оперативната памет (ОП). **Кодът** на инструкцията определя операцията, а **аргументите** – данните, с които да се извърши предписаната операция. ЦП извлича инструкцията и зададените в нея аргументи от ОП, стартира схемата за извършване на операцията, връща резултата в ОП и определя следващата инструкция. Системата от инструкции на компютъра и правилата за изпълнението им наричаме **машинен език**. Поредица от инструкции наричаме **програма**.

```
1000 СЪБЕРИ X Y
1001 СЪБЕРИ X 5
1002 УВЕЛИЧИ Y
1003 ПРЕМЕСТИ X Y
1004 КРАЙ
```

	...
X	1000
	...
Y	8
	...

Фиг. 6. Програма

```
ПРИ=0 X A
ПРИ>=0 X A
СКОЧИ A
```

Фиг. 7. Инструкции за преход

```
1000 СЪБЕРИ X Y
1001 ПРИ>=0 X 1003
1002 УМНОЖИ X -1
1003 ПОКАЖИ X
1004 КРАЙ
```

Фиг. 8.

На Фиг. 6 е показана програма от четири инструкции и част от ОП. За удобство, двоичните кодове са представени с по-разбираеми означения. След изпълнението на всяка от тези инструкции ЦП преминава към изпълнение на следващата инструкция.

Първата инструкция предписва да се съберат X и Y, а резултатът да се съхрани в X. В резултат, съдържанието на X ще стане 1008 и ЦП ще премине към втората инструкция. Тя прибавя 5 към съдържанието на X и то става 1013. Третата инструкция увеличава съдържането на Y с единица, а четвъртата – заменя съдържанието на X с това на Y. Какви ще бъдат стойностите на X и Y, след изпълнението на всяка от тези инструкции? Инструкцията **КРАЙ** е указание за процесора да преустанови работата на програмата.

Инструкциите се разполагат в ОП, както и данните. Така всяка инструкция получава *адрес* в паметта. Числото пред всяка инструкция на Фиг. 6 е нейният адрес. Всеки машинен език има *инструкции за разклоняване* или *преход*. На Фиг. 7 са показани примери на такива инструкции. Инструкцията с код **ПРИ=0** предписва да се провери съдържанието на първия аргумент и ако е 0, ЦП трябва да премине към изпълнение на инструкцията с адрес във втория аргумент. В противен случай, т.е. X не е нула, ЦП трябва да продължи със следващата инструкция на програмата в ОП. **Когато ЦП изпълни инструкция, която не е за преход, винаги продължава със следващата в ОП инструкция.**

Възможни са и по-сложни проверки, както в **ПРИ>=0**, която извършва прехода, когато стойността на първия аргумент е по-голяма или равна на 0. Инструкцията **СКОЧИ** предписва преходът да се извърши, без да се проверява условие – *безусловен преход*. На Фиг. 8 е показана програма, която намира и показва на екрана абсолютната стойност на сумата на числата, намиращи се в паметта на адреси X и Y. Проследете работата на програмата, ако X = 3 и Y = -7.

Машинният език е труден за усвояване и неудобен за всекидневна употреба. Затова в масовото производство на програми се използват *езици за програмиране*, които са по-близки до човешкия. Един от най-популярните езици за програмиране е езикът C (четем „си“). От него са произлезли различни по-модерни езици, включително и езикът C# (четем „си шарп“), с който ще се запознаем в уроците по информатика.

И така, компютърът е сложно устройство (*хардуер*), изпълняващо програми, които решават интересни за потребителя задачи за съхраняване, обработка и изобразяване на данни. Програмите изпълнявани от

В другата група програмите са независими една от друга и се наричат *инструментални програми* (utilities). Може да се каже, че инструменталните програми са приложни програми, които решават сериозни задачи, подпомагат работата предимно на квалифицирани специалисти в областта на информатиката и затова се считат за започнем разглеждането на един основен за информатиките *разглеждане на един основен за информатиките* *рираните среди за разработване на софтуер*.



специалисти в областта на информатиката и затова се считат за част от системния софтуер. В следващ урок ще започнем разглеждането на един основен за информатиката клас инструментални средства – *интегрираните среди за разработване на софтуер*.

1. Защо идеята на Жакард за програмно управление е повратен момент в историята на създаването на компютрите?
2. Кой за пръв път споменава за двоична математическа теория, оперираща само с две стойности?
3. На кого принадлежи идеята за построяване на изчислителни устройства на електро-механичен принцип?
4. Кой е авторът на първия проект за електронни устройства, извършващи математически изчисления?
5. Коя е първата изчислителна машина, построена изцяло в съответствие с принципите на фон Нойман?
- 6\*. Кои от принципите на фон Нойман биха могли да се променят в резултат на развитието на науката и технологиите?
7. Защо наричат ЦП „мозък“ на компютъра?
- 8\*. Напишете програма на машинен език, пресмятаща изразите:  
а.  $A + B + C$ ;                                  б.  $A + B.C$ ;                                  в.  $A.B + C.D$ ,  
ако стойностите на  $A$ ,  $B$ ,  $C$  и  $D$  се намират в едноименни полета на паметта, а стойността на израза се запише в поле от паметта, означено с  $X$ .
- 9\*. Напишете програма на машинен език, пресмятаща верността на условията:  
а.  $A + B + C < 0$ ;                                  б.  $A + B > C$ ;                                  в.  $A.B \neq C.D$ ,  
ако стойностите на  $A$ ,  $B$ ,  $C$  и  $D$  се намират в едноименни полета на паметта, а верността на условието – означена с 0 за неистина и 1 за истина – се записва в поле от паметта, означено с  $X$ .