Java and ARM

Java, over time, has become one of the most widely used programming languages.
J2ME is a runtime and a collection of APIs for software development dedicated to devices with limited resources.
The portability of Java, and all the benefits gained from using J2ME, had led Java soon to become a de-facto development standart for mobile devices.

On the other side, the main problem of ARM architecture is the limited computational power.
A solution that has been found was to make cpu natively execute Java Bytecode.
The following is an overview about technologies used in ARM Architectures to treat instructions.

Jazelle

Jazelle DBX (Direct Bytecode eXecution) allows some ARM processors to natively execute Java bytecode.
The target of this tecnology is to facilitate the execution on Java ME programs on mobile devices.
The BXJ (Branch and eXchange to Java) instruction attempts to switch to Jazelle state.
If allowed and successful, sets the 'J' bit in the CPSR; otherwise, it "falls through" and acts as a standard BX (Branch) instruction.

Thumb

To improve compiled code-density since 1994 arm processors have featured Thumb instruction set, which have their own state.
When in this state, the processor executes the Thumb instruction set, a compact 16-bit encoding for a subset of the ARM instruction set.
Most of the Thumb instructions are directly mapped to normal ARM instructions.

Thumb2

Both ARM and Thumb instructions sets were exdended.
A stated aim for Thumb-2 was to achieve code density similar to Thumb with performance similar to the ARM instruction set on 32-bit memory.
As you can see below, the same source code can so be compiled as ARM or Thumb2 code.
Recall that the Thumb MOV instruction has no bits to encode "EQ" or "NE", as MOVEQ and MOVNE are ARM instruction.

code	pseudo-code	ARM meaning	Thumb2 meaning
`CMP r0, r1`	--	comparison	comparison
`ITE EQ`	if r0 == r1	--	starts an If-Then-Else block
`MOVEQ r0, r2`	then r0 = r2	conditional	"Then" block
`MOVNE r0, r3`	else r0 = r3	conditional	"Else" block

ThumbEE

ThumbEE, also known as Jazelle RCT (Runtime Compilation Target), was announced in 2005.
It added a fourth processor mode, and made small changes to the Thumb-2 extended Thumb instruction set.
These changes make JIT compilers able to output smaller compiled code without impacting performance.
ThumbEE is a target for languages such as Java, C#, Perl, and Python, which all uses JIT compilers.

AArch64e

Every use of the ThumbEE instruction set have been deprecated at the and of 2011, as AArch64 have enstabilished.
AArch64 is a 64-bit architecture, main news are :

- New instruction set, A64
  - 31 general-purpose 64-bit registers
  - Instructions are still 32 bits long and mostly the same as A32
  - Most instructions can take 32-bit or 64-bit arguments
  - Addresses assumed to be 64-bit
- Advanced SIMD (NEON) enhanced
  - Has 32x128-bit registers (up from 16), also accessible via VFPv4
  - Supports double-precision floating point
  - Fully IEEE 754 compliant
  - AES encrypt/decrypt and SHA-1/SHA-2 hashing instructions also use these registers
- A new exception system
  - Fewer banked registers and modes

Kernel Linux 3.7 included pathes to support AArch64e.