GBA Memory Map

General Internal Memory
```
00000000-00003FFF   BIOS - System ROM         (16 KBytes)
00004000-01FFFFFF   Not used
02000000-0203FFFF   WRAM - On-board Work RAM  (256 KBytes) 2 Wait
02040000-02FFFFFF   Not used
03000000-03007FFF   WRAM - On-chip Work RAM   (32 KBytes)
03008000-03FFFFFF   Not used
04000000-040003FE   I/O Registers
04000400-04FFFFFF   Not used
```
Internal Display Memory
```
05000000-050003FF   BG/OBJ Palette RAM        (1 Kbyte)
05000400-05FFFFFF   Not used
06000000-06017FFF   VRAM - Video RAM          (96 KBytes)
06018000-06FFFFFF   Not used
07000000-070003FF   OAM - OBJ Attributes      (1 Kbyte)
07000400-07FFFFFF   Not used
```
External Memory (Game Pak)
```
08000000-09FFFFFF   Game Pak ROM/FlashROM (max 32MB) - Wait State 0
0A000000-0BFFFFFF   Game Pak ROM/FlashROM (max 32MB) - Wait State 1
0C000000-0DFFFFFF   Game Pak ROM/FlashROM (max 32MB) - Wait State 2
0E000000-0E00FFFF   Game Pak SRAM    (max 64 KBytes) - 8bit Bus width
0E010000-0FFFFFFF   Not used
```
Unused Memory Area
```
10000000-FFFFFFFF   Not used (upper 4bits of address bus unused)
```

Default WRAM Usage
By default, the 256 bytes at 03007F00h-03007FFFh in Work RAM are reserved for Interrupt vector, Interrupt Stack, and BIOS Call Stack. The remaining WRAM is free for whatever use (including User Stack, which is initially located at 03007F00h).

Address Bus Width and CPU Read/Write Access Widths
Shows the Bus-Width, supported read and write widths, and the clock cycles for 8/16/32bit accesses.

```
Region         Bus   Read      Write     Cycles
BIOS ROM       32    8/16/32   -         1/1/1
Work RAM 32K   32    8/16/32   8/16/32   1/1/1
I/O            32    8/16/32   8/16/32   1/1/1
OAM            32    8/16/32   16/32     1/1/1 *
Work RAM 256K  16    8/16/32   8/16/32   3/3/6 **
Palette RAM    16    8/16/32   16/32     1/1/2 *
VRAM           16    8/16/32   16/32     1/1/2 *
GamePak ROM    16    8/16/32   -         5/5/8 **/***
GamePak Flash  16    8/16/32   16/32     5/5/8 **/***
GamePak SRAM   8     8         8         5     **
```
Timing Notes:
```
*    Plus 1 cycle if GBA accesses video memory at the same time.
**   Default waitstate settings, see System Control chapter.
***  Separate timings for sequential, and non-sequential accesses.
One cycle equals approx. 59.59ns (ie. 16.78MHz clock).
```
All memory (except GamePak SRAM) can be accessed by 16bit and 32bit DMA.

GamePak Memory
Only DMA3 (and the CPU of course) may access GamePak ROM. GamePak SRAM can be accessed by the CPU only - restricted to bytewise 8bit transfers. The SRAM region is supposed for as external FLASH backup memory, or for battery-backed SRAM.
For details about configuration of GamePak Waitstates, see:
GBA System Control

VRAM, OAM, and Palette RAM Access
These memory regions can be accessed during H-Blank or V-Blank only (unless display is disabled by Forced Blank bit in DISPCNT register).
There is an additional restriction for OAM memory: Accesses during H-Blank are allowed only if 'H-Blank Interval Free' in DISPCNT is set (which'd reduce number of display-able OBJs though).
The CPU appears to be able to access VRAM/OAM/Palette at any time, a waitstate (one clock cycle) being inserted automatically in case that the display controller was accessing memory simultaneously. (Ie. unlike as in old 8bit gameboy, the data will not get lost.)

CPU Mode Performance
Note that the GamePak ROM bus is limited to 16bits, thus executing ARM instructions (32bit opcodes) from inside of GamePak ROM would result in a not so good performance. So, it'd be more recommended to use THUMB instruction (16bit opcodes) which'd allow each opcode to be read at once.
(ARM instructions can be used at best performance by copying code from GamePak ROM into internal Work RAM)

Data Format
Even though the ARM CPU itself would allow to select between Little-Endian and Big-Endian format by using an external circuit, in the GBA no such circuit exists, and the data format is always Little-Endian. That is, when accessing 16bit or 32bit data in memory, the least significant bits are stored in the first byte (smallest address), and the most significant bits in the last byte. (Ie. same as for 80x86 and Z80 CPUs.)