Index

A

Advanced Configuration and Power Interface (ACPI), 2
AMD Generic Encapsulated Software Architecture (AGESA), 59
Application programming interface (API)
  data structure, 36
  execution status, 45
  FspInitEntry, 42
  hard-coded constants, 38
  NotifyPhase, 43
  TempRamInit, 39

B

Binary Configuration Tool (BCT), 30
BIOS Writer's Guide (BWG), 6
Boot Device Selection (BDS), 141
Boot Setting File (BSF), 30, 179

C

Chromebook firmware
  advantage, 97
  boot log, 104–105
  Chrome OS security, 98–99
  coreboot, 98
    ARM, 101
    x86, 101
  crossystem, 115–117
  Depthcharge payload
    kernel security, 104
    lib payload, 102
    verified boot, 103–104
    verified boot flow, 101–102
  Developer Mode, 100
  embedded controller
    battery charging, 119
    keyboard controller, 119
    power sequencing, 118
    responsibilities, 118
    software sync, 120
    thermal management, 119
  event log, 105–106
  firmware image, 98
  flashrom utility, 114
  FMAP, 107
    read-write section, 108–109
    structure, 107
  FMAP BOOT_STUB section, 108
  GBB
    bitmaps, 110
    boot flags, 110–111
    HWID v3, 110
    information, 109
    public and private keys, 110
  gbb_utility, 114–115
  mosys utility, 117
  powerwash, 99
  Recovery Mode, 99
  TPM, 112
  updating process, 113
  verified boot, 99
  VPD, 112
Commercial off-the-shelf (COTS), 15
Coreboot, 19
  AGESA reference, 59
  binaries, 79
  CAR, 58
  CBFS
    cbfstool, 78
    organization, 77
    size, 79

Coreboot (*cont.*)
  chipset driver
    chipset UPD options, 90
    Depthcharge, 93
    ELF format, 92
    FILO, 93
    FSP driver, 90–91
    GRUB2, 93
    Intel FSP driver, 90
    iPXE, 93
    Kconfig, 91
    Key files, 90
    libpayload, 94
    memory test, 94
    SeaBIOS, 92
    TianoCore, 93
    U-Boot, 93
    xcompile, 92
  community, working, 64–65
  community organization, 61
  CPU architectures, 57
  device tree, 84–86
    chips, 83, 86
    CPU and PCI busses, 82
    keywords, 84
    operations, 87
    sconfig tool, 82
    state machine states, 88
    types, 83
    variables, 83
  etherboot project, 58
  FSF, 58
  git commit messages, 62
  Git Gerrit, 61
  Google, 59
  Hardwaremain state machine, 87
  Intel FSP
    payload, 82
    ramstage, 81–82
    reset vector and bootblock, 80
    romstage, 80–81
  libreboot, 60
  LinuxBIOS, 57, 59
  logo, 56
  mainboard directory, 88–89
  MinnowMax (*see* MinnowBoard MAX (MinnowMax))
  nonsource binaries, 65
  philosophy, 56
  ROMCC, 58
  Sage, 60
  sign-off procedure, 63–64
  stages, 76
  state machine callbacks, 88
  state machine states, 87
  statistics, 60
  supporting files, 77
  troubleshooting and debugging, 94–95
  web site, 61

D

Double Data Rate (DDR) technology, 34–35

E

EFI Developer Kit II (EDKII), 125
EFI Development Kit II (EDK II)
  build command, 156
  DEC, 153
  DSC, 155
  FDF, 156
  INF file, 158
  packages, 149
  packaging
    IntelFrameworkModulePkg, 149
    IntelFrameworkPkg, 148
    MdeModulePkg, 148
    MdePkg, 147
    UEFI PI boot flow, 148
  PCD
    build and source construction, 152
    build flow, 152
    flash image, 153
    INF, DEC, and DSC relationship, 151
    Syntax, 152
  supporting files, 150
Embedded engineers, 1
Extensible Firmware Interface (EFI)
  coreboot and UEFI PI, 122
  development flow, 121
  FSP, 123
  history, 121
  PEIMs, 122
  SCM, 123
  SiRC, 123
  UEFI/EDK II, 123

F

Firmware
  ACPI table, 2
  BIOS vendors, 11
  boot process, 2
  chip vendors, 11
  coreboot and EDK, 10
  definition, 1
  engineers, 1, 3
  flash storage device, 3
  FPGA-based emulators, 4
  Google and Facebook, 10
  hardware design, 2
  Intel® FSP, 7
  Intel Quark family products, 10
  Intel's Simics and AMD's SimNow, 3
  Internet of Things, 10
  open source stacks, 9
  PC BIOS stack, 8–9
  programming guides, 6
  RTOS codebase, 10
  RTOS stacks, 9
  SMM and SMI, 2
  software vendors, 11
  Tianocore source code, 11
Firmware stacks
  BIOS, 13
  COTS, 15
  IBV, 20
  in-house development, 23
  ISV, 20
  legacy BIOS, 17
  microkernel, 16
  one-size-fits-all solution, 15
  open source, 18
  PC architecture, 15
  proprietary, 19
  RTOS, 16
  UEFI Framework, 18
  virtual machine monitor, 15
Firmware Support Package (FSP), 123
  APIs, 124
  architecture, 124
  binary-enabling model, 143
  Capsule Flash Update
    boot flow, 134–135
    memory layout, 135–136
    recovery boot, 136–138
  components, 125
  consumer, 123
  definition, 123
  EDKII, 138
  evolution, 143
  Minnow and MinnowMax
    APIs, 141
    BDS, 141
    binary, 140–141
    infrastructure code, 140
    locations, 140
    PCDs, 142
    Platform.dsc, 142
    Platform.fdf, 141
    workflow, 141
  normal boot
    boot flow, 128
    data structure, 130–131
    memory layout, 129–130
  producer, 123
  S3 boot
    boot flow, 132
    memory layout, 132–133
    NV data passing, 133–134
  UEFI payload, 138–140
  UEFI PI, 124
  wrapper boot flow
    APIs, 126
    boot modes, 128
    EDKII BIOS, 126
    FspNotifiyDxe, 126–127
    FspNotifyApi, 126
    integration options, 127
    SEC integration, 126
FspInit, 45
FspInitEntry
  ContinuationFunc, 43
  NvsBufferPtr points, 42
  prototype, 42
  RtBufferPtr points, 42

G

General Public
    License (GPL), 29

H

Hard-coded constants, 38

I, J, K

Independent BIOS Vendor (IBV), 20
Independent Software Vendors (ISVs), 20
Intel® Firmware Support
  Package (Intel FSP), 7
  API
    FspInitEntry, 42
    NotifyPhase, 43
    TempRamInit, 39
  binary format, 31
  Boot Setting File (BSF), 30
  community effort, 176
  Coreboot (*see* Coreboot)
  customization, 47
  customization and production, 176
  downloading, 49
  ecosystem perspective, 173
  expanded folder, 28
  future aspects, 53
  hard-core open source, 175
  header files and sample code, 31
  integration and build process, 53
  locating APIs
    data structure, 36
    hard-coded constants, 38
  microcode patches, 52
  output
    API execution status, 45
    non-volatile storage HOB, 46
    sample code for
      parsing HOBs, 46
    temporary memory data HOB, 45
  philosophy, 27
  prebuilt silicon, 176
  relocation, 53
  RTOS, 174
  sample boot flow, 34
  shipping of binary file, 29
  vertical sectors, 173
Intel Quark System-on-Chip (SoC)
  build options, 165
  compiler options, 165
  fixed DRAM resource, 161
  full SMM core solution, 164
  hardware platform, 160
  memory usage
    optimization, 169–170
  RAM footprint optimization, 168
  reduce features, 163–164
  remove features, 162
  SEC ➤ DXE solution, 162
  simplified SMM solution, 165
  TinyQuarkOptimization, 167

L

Linux, 18

M

Memory Reference Code (MRC), 34
Microcode, 28, 52
Microkernel, 16
MinnowBoard MAX (MinnowMax)
  building, 73–74
  Chipset menu, 72
  commands, 74
  commit hooks, 69
  development branch, 69
  development directory, 67
  devices menu, 73
  environment, 66
  flashing
    devices, 75
    flash programmer, 75
    ROM image, 75–76
  hardware, 66
  Intel FSP, 67–68
  Menuconfig menu, 71
  platforms, 66
  source download, 68
  toolchains, 68–69
Model-specific register (MSR), 7
Module Development Environment
  (MDE) Package (MdePkg), 147
Multiple Independent Levels
  of Security (MILS), 174

N, O

Non-volatile storage HOB, 46
NotifyPhase, 45

P, Q

Platform Configuration
  Database (PCD), 150

R

Real-time operating systems (RTOS), 2, 16

S

System Management Interrupt (SMI), 2
System Management Mode (SMM), 2

T

Temporary memory data HOB, 45
TempRamInit, 39, 45
Tianocore, 11
Trusted Platform Module (TPM), 112

U

U-Boot community, 19
Unified Extensible Firmware Interface (UEFI)
- integration, Intel FSP (*see* Intel® Firmware Support Package (Intel FSP))
- firmware stacks, 18

UEFI Firmware Writer's Guide, 6
UEFI's Firmware Volume (FV) layout format, 31
Updatable Product Data (UPD), 47

V, W, X, Y, Z

Vital Product Data (VPD), 47–48