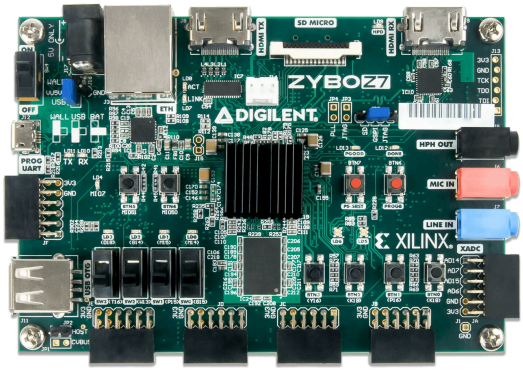
IronForge on Zybo Z7-20

IronForge runs Linux on both Cortex-A9 cores and drives a 1080p60 HDMI console from a 32-bit framebuffer in DDR. The platform includes stereo audio, Pcam 5C capture over MIPI CSI-2, 1080p30 hardware H.264 Baseline streaming over Gigabit Ethernet, 16 MiB of QSPI NOR flash, and FAT storage on microSD and USB. Network boot loads the kernel and FPGA design over TFTP and runs applications directly from the Windows NFS root. Standalone QSPI boot uses a bundled ramdisk.



Digilent Zybo Z7-20 · Part 410-351-20. Select the Z7-20 variant on the Digilent product page. Photo by Digilent.

Platform	Board and runtime
SoC and memory	Zynq-7000 `XC7Z020-1CLG400`, two Cortex-A9 processors and 1 GiB DDR3L
Common core	Linux 7.3.4, U-Boot 2026.10 and BusyBox 1.38.0
Ethernet	PS GEM0, RTL8211F Gigabit RGMII PHY, DHCP and TFTP boot
Display and input	IronVideo HDMI TX with 720p60 and engineering 1080p60 modes, PS UART1 at 115200 8N1 and USB HID support
Camera and audio	Optional two-lane Pcam 5C capture and SSM2603 stereo analog playback
Boot media	16 MiB QSPI NOR and native four-bit microSD. The network FIT starts a writable RAM root.

Local and remote operation

U-Boot displays a coordinate pattern after loading a known FPGA image. Linux then takes over with a DDR-backed DRM framebuffer. The serial console mirrors ordinary output to HDMI VT1. A USB keyboard can open a local shell on VT2 when the Linux USB host is enabled.

JTAG can initialize the PS and DDR and start U-Boot in RAM. A Linux affinity test ran on each core in turn. Linux reported 333.33 BogoMIPS per core. The CoreMark section gives measured CPU performance.

Download the Zybo Z7-20 QSPI demo

Choose the HDMI mode you want at power-on. Each 16 MiB image boots to a BusyBox shell and includes the FPGA design, Linux, the rotating cube and its stereo soundtrack. Ethernet and an SD card are optional.

HDMI factory default	Programming bundle	Raw flash image
1280 × 720 at 60 Hz	Download 720p60 ZIP	ironforge-zybo-z7-20-qspi-720p60.bin
1920 × 1080 at 60 Hz	Download 1080p60 ZIP	ironforge-zybo-z7-20-qspi-1080p60.bin

Download the ZIP for a first installation. It includes one `flash.bin`, checksums, USB programming helpers and instructions. Programming replaces all flash contents, including the saved U-Boot environment. Power off after programming, fit JP5 across the two center pins for QSPI, and power on with a monitor connected to HDMI TX.

At the shell

Run `ironforge-cube` to start the animation with 48 kHz stereo sound at HPH OUT. Both headphone channels start at −12 dB with HiFi playback enabled. Ctrl+C restores the console. The USB UART runs at 115200 8N1. A local USB keyboard can open an HDMI shell with Alt+F2 and Enter.

Dropbear starts independently of the shell. To log in remotely, run `ironforge-demo-ssh` at the physical console and paste a client Ed25519 public key. The server identity and client authorization live in RAM and expire at reboot.

Both downloads support both HDMI modes. The 720p60 serializer runs at 371.25 MHz. The 1080p60 engineering mode uses a 742.5 MHz serializer clock above the XC7Z020-1 speed-grade rating.

On the test board the 1080p60 factory image passed a full-chip readback, booted from SPI, and reached the shell with both audio controls at their factory settings. The operator confirmed the picture and soundtrack after launching the cube. Dropbear rejected an unenrolled client and accepted a freshly enrolled key with strict host-key verification.

The 720p60 factory image also passed full-chip readback. With Ethernet unplugged, a cold boot reached BusyBox with the expected audio settings. The manually launched cube presented 299 frames in 5.013 seconds with zero display underflows or read errors.

Processing system and FPGA architecture

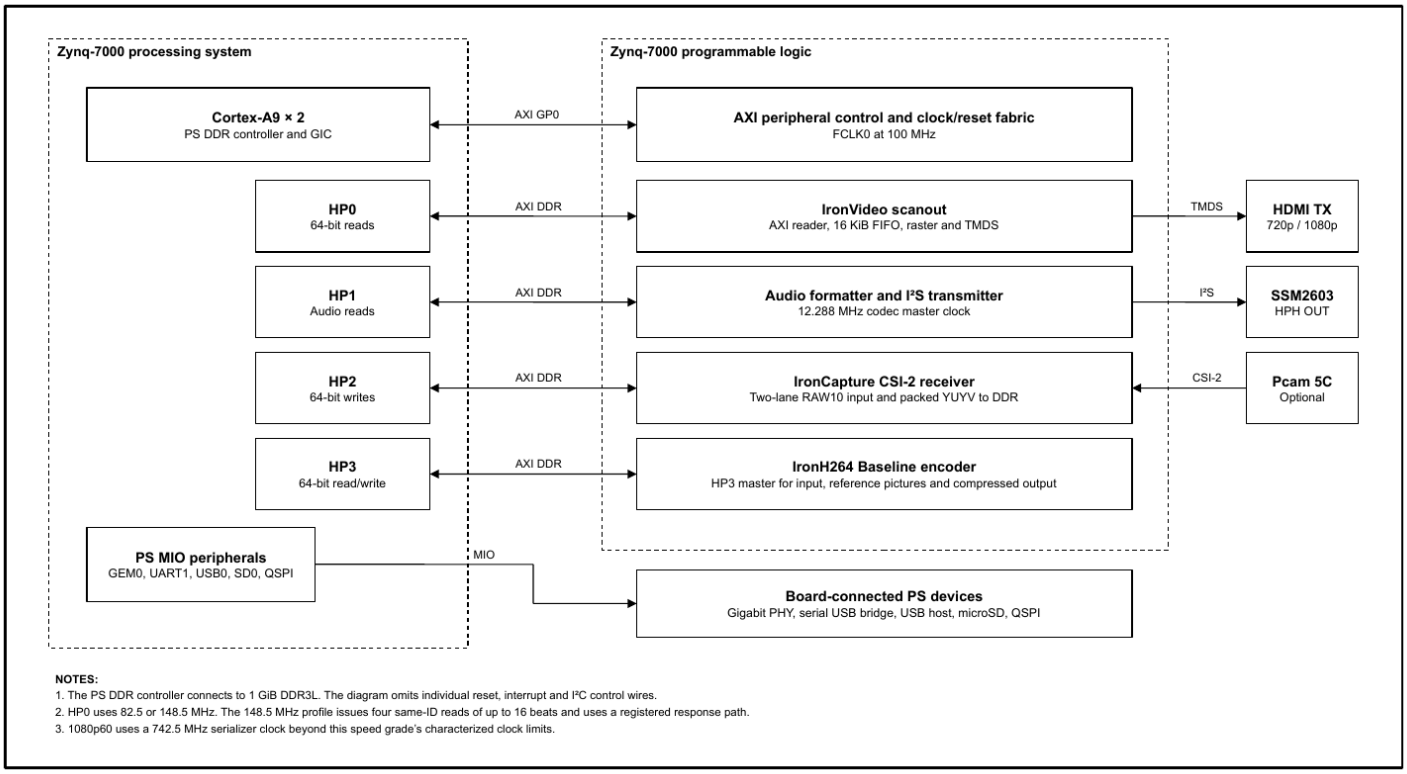


Figure 1. The Baseline build connects four FPGA bus masters to separate DDR ports. The native GEM, USB, SD, QSPI and UART remain in the Zynq PS.

Logic	Control and DDR path
IronVideo	`0x43000000`, GIC SPI 29. HP0 reads at 82.5 or 148.5 MHz over 64-bit AXI. The fast profile allows four same-ID reads of up to 16 beats and registers the AXI read-response path.
Audio	Formatter `0x40400000`, SPI 30, reads DDR on HP1. I²S `0x43c20000`, SPI 31, feeds the SSM2603 codec with a 12.288 MHz master clock. I²C control is at `0x40800000`.
Camera	IronCapture CSI-2 at `0x43c00000`, SPI 33. HP2 writes packed YUYV after two-lane RAW10 conversion. The camera build adds I²C, sensor enable and a 200 MHz IDELAY reference.
H.264	IronH264 at `0x43c40000`, GIC SPI 35. Its 64-bit HP3 master reads captured frames and accesses reference pictures and the compressed bitstream in DDR.

FCLK0 at 100 MHz supplies the control fabric and audio and camera AXI domains. IronVideo crosses DDR data into the raster domain through a 16 KiB FIFO. The camera-only 148.5 MHz AXI build closed setup at +0.414 ns and hold at +0.052 ns. Packaging checks the selected device tree against the hardware clock and address map.

HDMI console and DDR scanout

Mode	Pixel / serializer	Physical result
720p60	74.25 / 371.25 MHz	A monitor displayed Cortex-A9 DDR color bars. IronVideo counted 3,600 frames with zero underruns, AXI read errors or faults.
1080p60 RGB565	148.5 / 742.5 MHz	With HP0 at 82.5 MHz, the normal Linux console advanced 4,205 frames in 70.09 seconds with zero underruns, AXI read errors or faults.
1080p60 XRGB8888	148.5 / 742.5 MHz	At 148.5 MHz HP0 with four reads in flight, the normal Linux console advanced 4,204 frames in 70.07 seconds with zero underruns, AXI read errors or faults.

The XRGB8888 framebuffer uses 7,680 bytes per active row. IronVideo reported 1.158 GB/s peak DDR row fetch bandwidth. The slowest observed row took 1,220 pixel clocks against a 2,200-clock line budget. The FIFO stayed at or above 1,030 of 2,048 beats. The counter advanced by 4.36 billion 64-bit DDR beats after accounting for wrap. NFS was mounted and two serial-only markers appeared on VT1. The J12-powered measurement used temporary boot arguments that skipped the sensor probe and disabled Linux USB. Camera capture was idle.

The earlier single-read controller peaked at about 434 MB/s and recorded 1,892 underruns in a 63.05-second 32-bit test. The final reader overlaps addresses with responses while retaining response order on one AXI ID.

U-Boot also drove its internally generated 1080p60 coordinate pattern. The operator saw stable edges and its counter advanced 125 frames in 2.05 seconds. The pattern bypasses DDR.

The 742.5 MHz serializer clock exceeds the speed-grade −1 clock-network rating stated in AMD DS187. The AXI and 1080p pixel domains passed setup and hold, and the high-rate serializer reported a −0.320 ns pulse-width violation. The 1080p60 measurements apply to the tested board and display.

Rotating cube on HDMI

The `ironforge-cube` demo software-rasterizes a rotating 3×3 cube, composes in cached RAM and copies the changed rectangle into an active XRGB8888 DRM buffer. All six faces turn in sequence and then retrace their moves. IronVideo continues scanning DDR independently. The mute mode uses the monotonic clock. The original soundtrack is a four-voice chip tune, provided separately as a 29.09-second, 48 kHz stereo digital WAV reference. The live demo plays the score through HPH OUT.

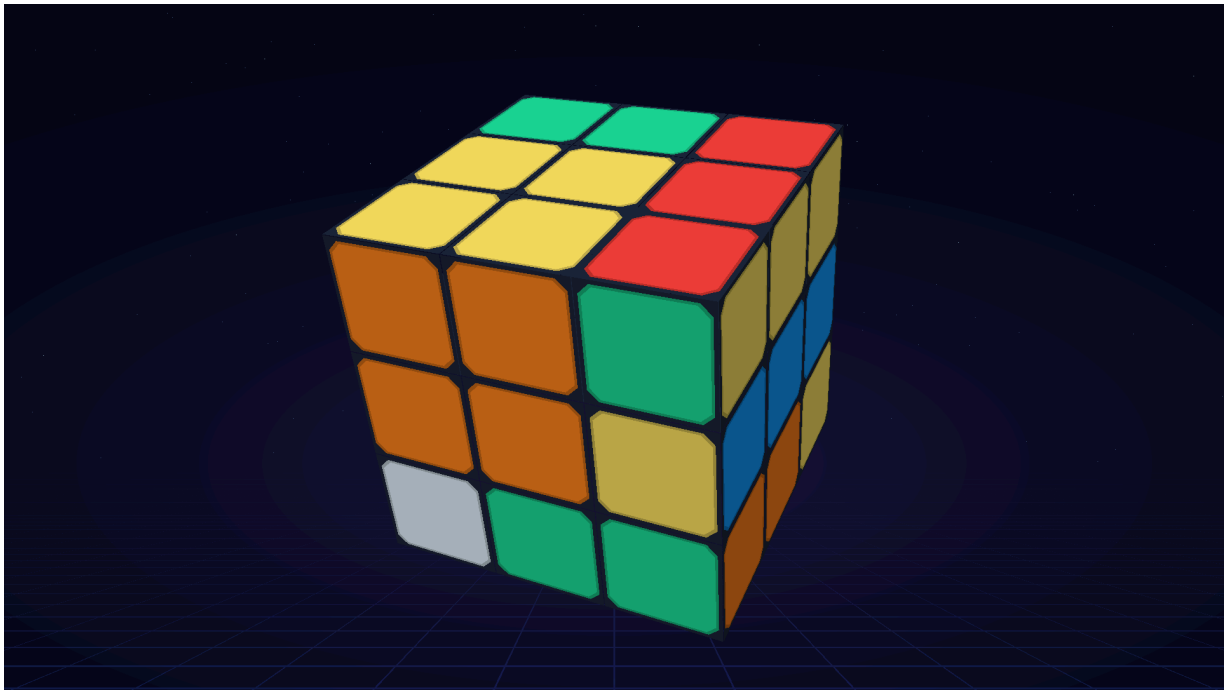
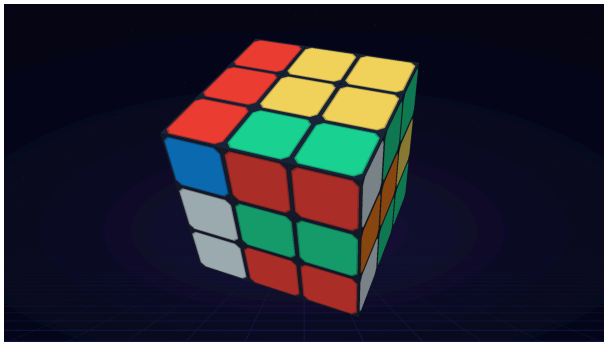


Figure 2. A native 1920 × 1080 capture of the active DRM buffer on Zybo.



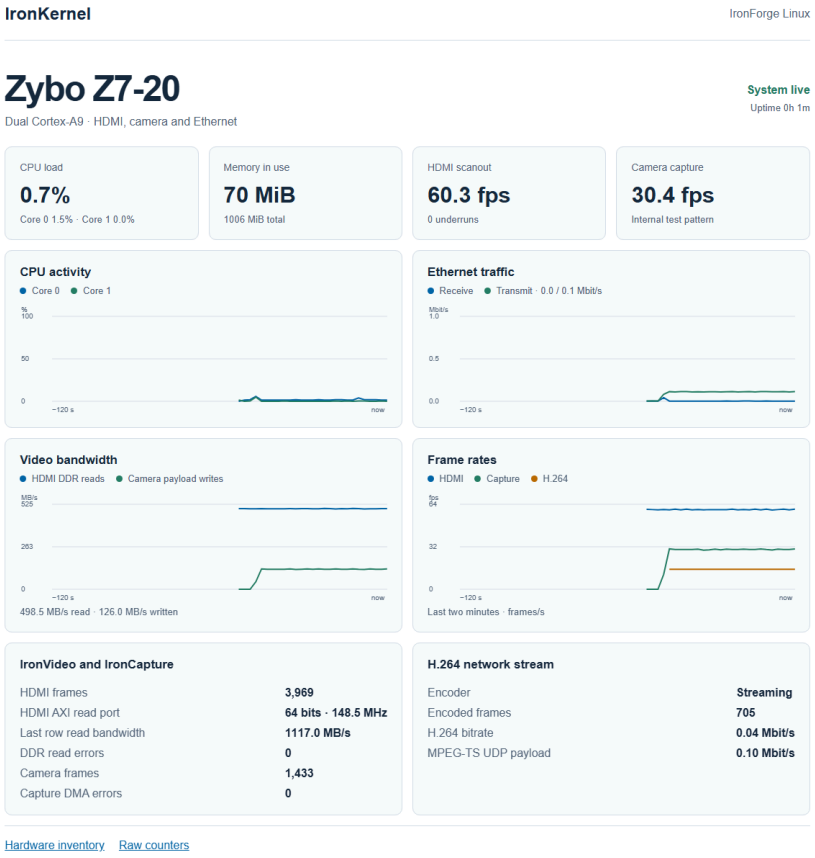


Figures 3 and 4. Two later poses from the same physical run.

With the ARM clock temporarily set to 333.333 MHz, the muted run presented 265 frames in 12.029 seconds (22.03 frames/s) while writing the three snapshots. A separate run presented 273 frames in 12.023 seconds (22.71 frames/s) without snapshot writes. During that run, IronVideo scanned 484 frames in 8.06 seconds (60.05 Hz), with zero underruns, AXI read errors or faults. Camera capture stayed idle and the PCM device stayed closed.

Live video dashboard and H.264 streaming

The onboard web server plots two minutes of per-core CPU load, Ethernet traffic, video bandwidth and frame rates. HDMI bandwidth counts accepted AXI read beats. Capture bandwidth counts the YUYV bytes in completed frames. The dashboard identifies the internal pattern when it supplies the stream.



Physical Pcam, external 5 V supply. IronCapture wrote 1920 × 1080 YUYV pictures to DDR. The retimed 121.85 MHz IronH264 Baseline encoder delivered **29.9997 pictures/s** over Gigabit Ethernet. FFmpeg decoded all 1,800 pictures. The longest interval between received picture timestamps was 33.722 ms.

Physical camera measurement	90 MHz encoder	121.85 MHz encoder
Decoded pictures / first-to-last timestamp span	1,800 / 76.567 s	1,800 / 59.967 s
Delivered pictures/s	23.50	30.00
H.264 / MPEG-TS UDP payload	14.63 / 15.04 Mb/s	20.45 / 21.02 Mb/s
Concurrent USB read / transport continuity errors	1 GiB / 0	1 GiB / 0
HDMI underruns / DDR read errors / capture DMA errors	0 / 0 / 0	0 / 0 / 0

HDMI scanned a 32-bit 1080p60 framebuffer throughout both measurements. In a separate capture-only test, the Pcam delivered 600 consecutive frames.

CPU and Ethernet measurements

CoreMark on the Cortex-A9

The board ran EEMBC CoreMark as a static Linux executable using the upstream POSIX port, GCC 13.3.1, `-O2`, Cortex-A9 tuning and the soft-float ABI. The PS CPU clock was 666.667 MHz. Each invocation used one CPU context and 30,000 iterations.

CoreMark parameter set	Elapsed	Iterations/s	Result
2K performance	12.844 s	2,335.72	Standard CRCs passed, exit 0
2K validation	12.899 s	2,325.76	Standard CRCs passed, exit 0

The first row is the single-context CoreMark score. Both runs exceeded EEMBC's 10-second reporting threshold.

Ethernet link and throughput

The native PS GEM0 negotiated 1 Gbit/s full duplex. Both endpoints ran iperf3 3.18 on the same wired LAN, with Windows as client and Zybo as a one-shot server. Reverse tests sent traffic from Zybo. The PS CPU clock was 666.667 MHz. All tests used one stream, with 12-second TCP runs and 10-second UDP runs carrying 1,400-byte datagrams. HDMI scanned out at 1080p60 XRGB8888 with zero additional underruns, read errors or faults. Camera and audio were idle. NFS stayed mounted read-only without concurrent benchmark traffic.

Protocol and direction	UDP target (Mbit/s)	Sent (Mbit/s)	Received (Mbit/s)	UDP loss	Jitter (ms)
TCP Windows → Zybo	—	760.38	758.97	—	—
TCP Zybo → Windows	—	693.03	692.53	—	—
UDP Windows → Zybo	100	99.86	98.06	1.79%	0.083
UDP Windows → Zybo	300	299.65	280.40	6.41%	0.150
UDP Windows → Zybo	600	599.76	591.35	1.32%	0.027
UDP Windows → Zybo	900	899.24	702.06	21.85%	0.025
UDP Zybo → Windows	100	100.00	100.01	0.00%	0.022
UDP Zybo → Windows	300	198.88	198.91	0.00%	0.035
UDP Zybo → Windows	600	373.88	373.93	0.00%	0.034
UDP Zybo → Windows	900	374.81	374.87	0.00%	0.032

Zybo's reverse UDP sender delivered about 375 Mbit/s at the higher targets. Windows received every datagram the sender reported for those runs.

Boot media and persistence

Network boot and NFS

For network development, U-Boot fetches `system-nfs.itb` over TFTP. Linux mounts the ARM filesystem exported by Windows directly at `/` over NFSv3/TCP. For standalone QSPI boot, `system.itb` bundles a compressed LZ4 initramfs alongside the kernel, device tree and FPGA bitstream.

Onboard NOR

QSPI region	Start	Size
Zynq SPL `BOOT.BIN`	`0x000000`	1 MiB
U-Boot `u-boot.img`	`0x100000`	1 MiB
Linux and FPGA FIT `system.itb`	`0x200000`	12 MiB
Primary / redundant U-Boot environment	`0xe00000` / `0xe40000`	128 KiB each

The physical Winbond W25Q128JV identifies as JEDEC `ef7018`. JP5 selects QSPI or SD boot. The flash packager emits individual payloads for selective updates and a 16 MiB initialization image. Selective updates preserve the environment slots. The tested board's factory EUI-48, `00:18:3e:05:07:3d`, was read from security register address `0x001000`. The final U-Boot automatically imports that address when `ethaddr` is missing and preserves a valid administrator assignment. Clearing the value in RAM, restoring the factory value and retaining a valid override all passed on the board. An explicit `saveenv` persisted the factory value across reset.

microSD

The Rev D.1 socket J4 receives fixed 3.3 V. A TXS02612 level shifter connects the 1.8 V PS MIO bank to the socket. MIO40–45 carry clock, command and four data lanes. MIO47 detects the card. The SDHCI power setting is the host's requested value, while the socket rail requires a measurement.

The original card timed out on ACMD13. With the replacement SDHC card, U-Boot identified a 29.7 GiB ADLCD and negotiated four-bit transfers at 50 MHz. Linux mounted the FAT partition read-only and completed a 256 MiB sequential read. Two independent reads of the first 4 MiB of the partition produced the same SHA-256.

Running Linux directly from the Windows NFS root

On the development PC, WinNFSd exports `C:\IronKernel\IronForge\rootfs` as `/ironforge`. The Zybo reads its applications and configuration from that tree. An update to `rootfs/usr/bin/ironforge-cube` reaches the board at `/usr/bin/ironforge-cube`. Relaunch the demo to use the updated executable. Linux keeps temporary runtime files in memory under `/run` and `/tmp`.

Stage	Observed on the Zybo
U-Boot	DHCP assigned `192.168.68.112`. TFTP transferred the 6,903,824-byte `system-nfs.itb` from `192.168.68.61`. The FIT contained the kernel, device tree and FPGA bitstream.
HDMI	IronVideo started its 1080p60 pattern before the kernel handoff.
Linux	The kernel mounted `192.168.68.61:/ironforge` read-only at `/` and executed `/init` from the export.
Application update	A host-side replacement of a test executable produced new output on its next invocation in the same BusyBox shell. The installed cube also matched the SHA-256 of the ARM build.

The following lines are excerpts from the serial console. The USB host and camera probe were disabled for this USB-powered boot.

```
Bytes transferred = 6903824 (695810 hex)
HDMI: IronVideo 1080p60 pattern active (3 frames advanced)
VFS: Mounted root (nfs filesystem) readonly on device 0:17.
Run /init as init process
```

Cold boot trace of the standalone image over TFTP

The board was switched off and back on with JP5 set to JTAG. The host captured COM6 at 115200 baud, 8N1, before power was removed. COM6 disconnected and returned. After power-on, XSDB read boot mode `0x00000000`, reset status `0x00400002` and CPU0 at `0xffffff28` in BootROM. BootROM waited silently for JTAG.

This earlier cold-boot capture exercised the standalone `system.itb` image over TFTP. The host reset the processing system, initialized DDR, programmed the FPGA and started U-Boot in RAM. Linux unpacked the bundled ramdisk. At the BusyBox shell, the host's NFS export was mounted read-only. The ARM ran at 666.7 MHz with the camera probe disabled.

```
Zynq> dhcp
DHCP client bound to address 192.168.68.112 (20 ms)
Zynq> setenv serverip 192.168.68.61
Zynq> tftpboot 0x10000000 system.itb
Using ethernet@e000b000 device
TFTP from server 192.168.68.61; our IP address is 192.168.68.112
Filename 'system.itb'.
Load address: 0x10000000
Loading: #################################################  8.1 MiB
     6.4 MiB/s
done
Bytes transferred = 8450736 (80f2b0 hex)
```

Serial console. This view of the recorded UART output shows U-Boot receiving all 8,450,736 bytes of `system.itb`.

```
# uname -sr
Linux 7.3.4
# cat /sys/class/net/eth0/speed
1000
# df -T /mnt/ironforge
Filesystem           Type       1024-blocks    Used Available Use% Mounted on
192.168.68.61:/ironforge
                     nfs        998824960 802819040 196005920  80% /mnt/ironforge
# ls -d /mnt/ironforge/bin /mnt/ironforge/usr
/mnt/ironforge/usr  /mnt/ironforge/bin
# _
```

HDMI console. This crop from the live 1920 × 1080 framebuffer shows the gigabit link and a directory read from the NFS mount. The following pages reproduce the complete UART trace in order, from the first U-Boot banner through the shell and mount verification.

U-Boot startup and TFTP

```
U-Boot 2026.10 (Jan 01 2025 - 00:00:00 +0000)

CPU:   Zynq 7z020
Silicon: v3.1
Model: Digilent Zybo Z7 board
DRAM:  ECC disabled 1 GiB
Core:  16 devices, 15 uclasses, devicetree: separate
MMC:   mmc@e0100000: 0
Loading Environment from SPIFlash...
SF: Detected w25q128jv with page size 256 Bytes, erase size 64 KiB, total 16 MiB
Loading Environment from SPIFlash... OK
In:    serial@e0001000
Out:   serial@e0001000
Err:   serial@e0001000
Net:
ZYNQ GEM: e000b000, mdio bus e000b000, phyaddr 0, interface rgmii-id
eth0: ethernet@e000b000
HDMI: IronVideo 1080p60 pattern active (3 frames advanced)
Zynq> setenv bootargs console=tty0 console=ttyPS0,115200 rdinit=/init initcall_blacklist=ov5640_i2c_driver_init
Zynq> setenv autoload no
Zynq> dhcp
DHCP client bound to address 192.168.68.112 (20 ms)
Zynq> setenv serverip 192.168.68.61
Zynq> tftpboot 0x10000000 system.itb
Using ethernet@e000b000 device
TFTP from server 192.168.68.61; our IP address is 192.168.68.112
Filename 'system.itb'.
Load address: 0x10000000
Loading: #################################################  8.1 MiB
     6.4 MiB/s
done
Bytes transferred = 8450736 (80f2b0 hex)
```

FPGA programming and kernel verification

```
Zynq> zybo_hdmi fit 0x10000000 ${filesize} && bootm 0x10000000
sha256+   design filename =
"system_wrapper;COMPRESS=TRUE;UserID=0XFFFFFFFF;Version=2026.1;SW_CRC=4b54f029;SW_HDR_SHA256=78d91f4cd3927ce1f3
6e476a06767d9e43342f78cc59f590fcb21efe10c99433"
  part number = "7z020clg400"
  date = "2026/11/20"
  time = "09:44:13"
  bytes in bitstream = 2131500
zynq_align_dma_buffer: Align buffer at 3ed0011b to 3ed00100(swap 1)
INFO:post config was not run, please run manually if needed
HDMI: IronVideo 1080p60 pattern active (3 frames advanced)
## Loading kernel (any) from FIT Image at 10000000 ...
   Using 'zybo' configuration
   Trying 'kernel' kernel subimage
     Description:  kernel
     Created:      1970-01-01   0:00:00 UTC
     Type:         Kernel Image
     Compression:  uncompressed
     Data Start:   0x100000bc
     Data Size:    4320352 Bytes = 4.1 MiB
     Architecture: ARM
     OS:           Linux
     Load Address: 0x02000000
     Entry Point:  0x02000000
     Hash algo:    sha256
     Hash value:   d722cb6228d65977bc2a64bd6c8d212b2573222ded6e02b7eef33aa4b0eb2311
   Verifying Hash Integrity ... sha256+ OK
```

Initramfs, device tree and kernel handoff

```
## Loading ramdisk (any) from FIT Image at 10000000 ...
   Using 'zybo' configuration
   Trying 'ramdisk' ramdisk subimage
     Description:  ramdisk
     Created:      1970-01-01   0:00:00 UTC
     Type:         RAMDisk Image
     Compression:  uncompressed
     Data Start:   0x10422f98
     Data Size:    1980867 Bytes = 1.9 MiB
     Architecture: ARM
     OS:           Linux
     Load Address: unavailable
     Entry Point:  unavailable
     Hash algo:    sha256
     Hash value:   873fec4253cef55fa9f4043abda80ab3b053f361f5badd18dfea550b23105d8b
   Verifying Hash Integrity ... sha256+ OK
## Loading fdt (any) from FIT Image at 10000000 ...
   Using 'zybo' configuration
   Trying 'fdt' fdt subimage
     Description:  fdt
     Created:      1970-01-01   0:00:00 UTC
     Type:         Flat Device Tree
     Compression:  uncompressed
     Data Start:   0x1041ee00
     Data Size:    16608 Bytes = 16.2 KiB
     Architecture: ARM
     Hash algo:    sha256
     Hash value:   34ac8b6e762dde1f51dcf2a20755c16b4ae1859815a11da9aa5e424589c3e377
   Verifying Hash Integrity ... sha256+ OK
   Booting using the fdt blob at 0x1041ee00
Working FDT set to 1041ee00
   Loading Kernel Image to 2000000
   Loading Ramdisk to 2fe1c000, end 2ffff9c3 ... OK
   Loading Device Tree to 2fe14000, end 2fe1b0df ... OK
Working FDT set to 2fe14000

Starting kernel ...
```

Linux memory and processor initialization

```
Booting Linux on physical CPU 0x0
Linux version 7.3.4 (ironforge@windows) (arm-none-eabi-gcc.exe (Arm GNU Toolchain 13.3.Rel1 (Build arm-13.24))
13.3.1 20240614, GNU ld (Arm GNU Toolchain 13.3.Rel1 (Build arm-13.24)) 2.42.0.20240614) #2 SMP Fri Nov 20
03:16:00 UTC 2026
CPU: ARMv7 Processor [413fc090] revision 0 (ARMv7), cr=18c5387d
CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing instruction cache
OF: fdt: Machine model: Digilent Zybo Z7-20
Memory policy: Data cache writealloc
cma: Reserved 64 MiB at 0x3c000000
OF: reserved mem: Reserved memory: No reserved-memory node in the DT
Zone ranges:
  Normal   [mem 0x0000000000000000-0x000000002fffffff]
  HighMem  [mem 0x0000000030000000-0x000000003fffffff]
Movable zone start for each node
Early memory node ranges
  node   0: [mem 0x0000000000000000-0x000000003fffffff]
Initmem setup node 0 [mem 0x0000000000000000-0x000000003fffffff]
percpu: Embedded 9 pages/cpu s15756 r0 d21108 u36864
Kernel command line: console=tty0 console=ttyPS0,115200 rdinit=/init initcall_blacklist=ov5640_i2c_driver_init
video=DVI-D-1:1920x1080-32@60
printk: log buffer data + meta data: 131072 + 409600 = 540672 bytes
Dentry cache hash table entries: 131072 (order: 7, 524288 bytes, linear)
Inode-cache hash table entries: 65536 (order: 6, 262144 bytes, linear)
Built 1 zonelists, mobility grouping on.  Total pages: 262144
mem auto-init: stack:all(zero), heap alloc:off, heap free:off
SLUB: HWalign=32, Order=0-3, MinObjects=0, CPUs=2, Nodes=1
rcu: Hierarchical RCU implementation.
rcu: RCU calculated value of scheduler-enlistment delay is 10 jiffies.
NR_IRQS: 16, nr_irqs: 16, preallocated irqs: 16
slcr mapped to (ptrval)
L2C: platform modifies aux control register: 0x02060000 -> 0x02460000
L2C: DT/platform modifies aux control register: 0x02060000 -> 0x02460000
L2C-310 enabling early BRESP for Cortex-A9
L2C-310 full line of zeros enabled for Cortex-A9
L2C-310 dynamic clock gating enabled, standby mode enabled
L2C-310 cache controller enabled, 8 ways, 512 kB
L2C-310: CACHE_ID 0x410000c8, AUX_CTRL 0x46460001
rcu: srcu_init: Setting srcu_struct sizes based on contention.
clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns: 19112604462750000 ns
zynq_clock_init: clkc starts at (ptrval)
Zynq clock init
sched_clock: 64 bits at 167MHz, resolution 6ns, wraps every 4398046511103ns
clocksource: arm_global_timer: mask: 0xffffffffffffffff max_cycles: 0x26703d7dd8, max_idle_ns: 440795208065 ns
Switching to timer-based delay loop, resolution 6ns
Console: colour dummy device 80x30
printk: legacy console [tty0] enabled
Calibrating delay loop (skipped), value calculated using timer frequency.. 333.33 BogoMIPS (lpj=1666666)
CPU: Testing write buffer coherency: ok
CPU0: Spectre v2: using BPIALL workaround
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 2048 (order: 1, 8192 bytes, linear)
Mountpoint-cache hash table entries: 2048 (order: 1, 8192 bytes, linear)
VFS: Finished mounting rootfs on nullfs
CPU0: thread -1, cpu 0, socket 0, mpidr 80000000
Setting up static identity map for 0x100000 - 0x100060
rcu: Hierarchical SRCU implementation.
rcu:    Max phase no-delay instances is 1000.
```

Secondary CPU, buses and network stack

```
smp: Bringing up secondary CPUs ...
CPU1: thread -1, cpu 1, socket 0, mpidr 80000001
CPU1: Spectre v2: using BPIALL workaround
smp: Brought up 1 node, 2 CPUs
SMP: Total of 2 processors activated (666.66 BogoMIPS).
CPU: All CPU(s) started in SVC mode.
Memory: 959152K/1048576K available (6144K kernel code, 541K rwdata, 1556K rodata, 1024K init, 225K bss, 21564K
reserved, 65536K cma-reserved, 196608K highmem)
devtmpfs: initialized
VFP support v0.3: implementor 41 architecture 3 part 30 variant 9 rev 4
posixtimers hash table entries: 1024 (order: 1, 8192 bytes, linear)
futex hash table entries: 512 (32768 bytes on 1 NUMA nodes, total 32 KiB, linear).
NET: Registered PF_NETLINK/PF_ROUTE protocol family
DMA: preallocated 256 KiB pool for atomic coherent allocations
/replicator: Fixed dependency cycle(s) with /axi/etb@f8801000
/axi/etb@f8801000: Fixed dependency cycle(s) with /replicator
/replicator: Fixed dependency cycle(s) with /axi/tpiu@f8803000
/axi/tpiu@f8803000: Fixed dependency cycle(s) with /replicator
/axi/ptm@f889d000: Fixed dependency cycle(s) with /axi/funnel@f8804000
/axi/funnel@f8804000: Fixed dependency cycle(s) with /axi/ptm@f889d000
/axi/ptm@f889c000: Fixed dependency cycle(s) with /axi/funnel@f8804000
/axi/funnel@f8804000: Fixed dependency cycle(s) with /axi/ptm@f889c000
/replicator: Fixed dependency cycle(s) with /axi/funnel@f8804000
/axi/funnel@f8804000: Fixed dependency cycle(s) with /replicator
/amba-pl/video-capture@43c00000: Fixed dependency cycle(s) with /amba-pl/i2c@40810000/camera@3c
/amba-pl/i2c@40810000/camera@3c: Fixed dependency cycle(s) with /amba-pl/video-capture@43c00000
e0001000.serial: ttyPS0 MMIO32:0xe0001000 (irq = 26, base_baud = 6249999) is a xuartps
printk: legacy console [ttyPS0] enabled
SCSI subsystem initialized
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
mc: Linux media interface: v0.10
videodev: Linux video capture interface: v2.00
pps_core: LinuxPPS API ver. 1 registered
pps_core: Software ver. 5.3.6 - Copyright 2005-2007 Rodolfo Giometti <giometti@linux.it>
PTP clock support registered
Advanced Linux Sound Architecture Driver Initialized.
clocksource: Switched to clocksource arm_global_timer
NET: Registered PF_INET protocol family
IP idents hash table entries: 16384 (order: 5, 131072 bytes, linear)
tcp_listen_portaddr_hash hash table entries: 512 (order: 0, 4096 bytes, linear)
Table-perturb hash table entries: 65536 (order: 6, 262144 bytes, linear)
TCP established hash table entries: 8192 (order: 3, 32768 bytes, linear)
TCP bind hash table entries: 8192 (order: 5, 131072 bytes, linear)
TCP: Hash tables configured (established 8192 bind 8192)
UDP hash table entries: 512 (order: 3, 28672 bytes, linear)
NET: Registered PF_UNIX/PF_LOCAL protocol family
RPC: Registered named UNIX socket transport module.
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
RPC: Registered tcp-with-tls transport module.
RPC: Registered tcp NFSv4.1 backchannel transport module.
```

Display, flash and peripheral drivers

```
Unpacking initramfs...
workingset: timestamp_bits=30 (anon: 25) max_order=18 bucket_order=0 (anon: 0)
Block layer SCSI generic (bsg) driver version 0.4 loaded (major 249)
io scheduler mq-deadline registered
io scheduler kyber registered
io scheduler bfq registered
zynq-pinctrl 700.pinctrl: zynq pinctrl initialized
Freeing initrd memory: 1936K
[drm] Initialized ironvideo 1.0.0 for 43000000.video on minor 0
Console: switching to colour frame buffer device 240x67
ironvideo 43000000.video: [drm] fb0: ironvideodrmfb frame buffer device
spi-nor spi0.0: supply vcc not found, using dummy regulator
7 fixed-partitions partitions found on MTD device spi0.0
Creating 7 MTD partitions on "spi0.0":
0x000000000000-0x000000100000 : "BOOT.BIN"
0x000000100000-0x000000200000 : "u-boot.img"
0x000000200000-0x000000e00000 : "system.itb"
0x000000e00000-0x000000e20000 : "u-boot-env"
0x000000e20000-0x000000e40000 : "reserved-env-gap"
0x000000e40000-0x000000e60000 : "u-boot-env-redund"
0x000000e60000-0x000001000000 : "reserved-tail"
macb e000b000.ethernet eth0: Cadence GEM rev 0x00020118 at 0xe000b000 irq 29 (00:18:3e:05:07:3d)
usbcore: registered new interface driver uas
usbcore: registered new interface driver usb-storage
i2c_dev: i2c /dev entries driver
clocksource: ttc_clocksource: mask: 0xffff max_cycles: 0xffff, max_idle_ns: 537538477 ns
timer #0 at (ptrval), irq=34
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver
xlnx_i2s 43c20000.i2s: xlnx_i2s_playback DAI registered
NET: Registered PF_PACKET protocol family
Registering SWP/SWPB emulation handler
mmc0: SDHCI controller on e0100000.mmc [e0100000.mmc] using ADMA
usb_phy_generic phy0: dummy supplies not allowed for exclusive requests (id=vbus)
ci_hdrc ci_hdrc.0: EHCI Host Controller
ci_hdrc ci_hdrc.0: new USB bus registered, assigned bus number 1
ci_hdrc ci_hdrc.0: USB 2.0 started, EHCI 1.00
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 1 port detected
clk: Disabling unused clocks
ALSA device list:
  #0: Zybo Z7-20
Freeing unused kernel image (initmem) memory: 1024K
```

BusyBox shell and read-only NFS mount

```
Run /init as init process
macb e000b000.ethernet eth0: PHY [e000b000.ethernet-ffffffff:00] driver [RTL8211F Gigabit Ethernet] (irq=POLL)
macb e000b000.ethernet eth0: configuring for phy/rgmii-id link mode
udhcpc: started, v1.38.0

IronKernel.com | IronForge ARM bring-up
Hardware acceptance required. Dashboard: http://<DHCP address>/
init started: BusyBox v1.38.0 (2025-01-01 00:00:00 UTC)
starting pid 64, tty '': '/bin/true'
starting pid 66, tty '/dev/ttyPS0': '-/bin/sh'


BusyBox v1.38.0 (2025-01-01 00:00:00 UTC) built-in shell (ash)

# udhcpc: broadcasting discover
mmc0: new high speed SDHC card at address aaaa
mmcblk0: mmc0:aaaa ADLCD 29.7 GiB
 mmcblk0: p1
usb 1-1: new high-speed USB device number 2 using ci_hdrc
usb-storage 1-1:1.0: USB Mass Storage device detected
scsi host0: usb-storage 1-1:1.0
scsi 0:0:0:0: Direct-Access     SanDisk  Cruzer Micro     8.02 PQ: 0 ANSI: 0 CCS
sd 0:0:0:0: [sda] 3907583 512-byte logical blocks: (2.00 GB/1.86 GiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] No Caching mode page found
sd 0:0:0:0: [sda] Assuming drive cache: write through
 sda: sda1
sd 0:0:0:0: [sda] Attached SCSI removable disk
macb e000b000.ethernet eth0: Link is Up - 1Gbps/Full - flow control off
udhcpc: broadcasting discover
udhcpc: broadcasting select for 192.168.68.112, server 192.168.68.1
udhcpc: lease of 192.168.68.112 obtained from 192.168.68.1, lease time 7200
uname -sr
Linux 7.3.4
# ip -4 addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    inet 192.168.68.112/22 brd 192.168.71.255 scope global eth0
       valid_lft forever preferred_lft forever
# mkdir -p /mnt/ironforge
# mount -t nfs -o ro,vers=3,nolock,tcp 192.168.68.61:/ironforge /mnt/ironforge
# grep ' /mnt/ironforge ' /proc/mounts
192.168.68.61:/ironforge /mnt/ironforge nfs
ro,relatime,vers=3,rsize=32768,wsize=32768,namlen=255,hard,nolock,fatal_neterrors=none,proto=tcp,timeo=600,retr
ans=2,sec=sys,mountaddr=192.168.68.61,mountvers=3,mountproto=tcp,local_lock=all,addr=192.168.68.61 0 0
# ls -d /mnt/ironforge/bin /mnt/ironforge/usr
/mnt/ironforge/usr  /mnt/ironforge/bin
# pcam-avdd: disabling
pcam-dovdd: disabling
pcam-dvdd: disabling
```

Board validation and native builds

Power and reset

The J12 USB-powered board intermittently returned to BootROM. JTAG read `REBOOT_STATUS=0x00400002`, indicating a power-on reset. A physical camera run reached stream-on before resetting. An audio-enabled cube run and some camera-idle boots also reset. JTAG confirmed the same reset status after the cube run. With the higher-power USB connection, a 666 MHz run with audio, HDMI and a concurrent SD read also returned to BootROM with the same reset flag. The input-rail voltage remains unmeasured. Digilent specifies a 0.75 A board limit on J12 and recommends a center-positive 4.5–5.5 V supply rated for at least 2.5 A at J17 with JP6 at WALL for demanding loads, as described in the Zybo Z7 reference manual.

Area	Verified behavior and scope
CPU and HDMI	Two CPUs passed affinity checks, CoreMark validated both parameter sets, and HDMI tests exercised 720p60 and 1080p60 DDR scanout.
Camera	With a continuous CSI-2 clock, the Pcam 5C delivered 120/120 frames at 720p30 and 300/300 at 1080p30. The 1080p60 HDMI console recorded zero underruns. NFS was mounted and the Linux USB host was disabled for this test.
Audio and HDMI	At 666.7 MHz the cube rendered 6,426 frames in 180.013 s at 1080p60. The owner heard the 48 kHz stereo soundtrack at HPH OUT. The run included a 256 MiB SD read and reported zero audio or video underruns.
USB host	With JP2 fitted, Linux detected a 2 GB SanDisk Cruzer Micro on J11 at 480 Mb/s, mounted its FAT partition read-only and completed a 16 MiB read.
Storage and networking	Read-only microSD, QSPI identification, factory MAC readout and TFTP FIT boot passed. Linux mounted the Windows export at `/` and ran an updated executable directly from it.

Native Windows workflow

PowerShell and MSYS drive the pinned ARM32 toolchain and Vivado. The Baseline build owns its Vivado project under the Zybo platform. The display-only variant omits the camera MMIO devices. Both use the same canonical source components.

```
.\ironforge.cmd --platform zybo-z7-20 --zybo-video-axi-hz 148500000 hardware --jobs 2
.\ironforge.cmd --platform zybo-z7-20 software all
.\ironforge.cmd --platform zybo-z7-20 software coremark
.\ironforge.cmd --platform zybo-z7-20 software iperf3
.\ironforge.cmd --platform zybo-z7-20 --zybo-video-axi-hz 148500000 package
```

Contact IronKernel

Adapt the Zybo Z7-20 camera-to-display design with IronKernel. We can integrate IronCapture and IronVideo on Zynq, bring up U-Boot and Linux, and develop custom camera or display drivers.

To discuss integration, reference designs or custom development, send your target board and project goal.

https://ironkernel.com/ · support@ironkernel.com