Booting linux on qemu-mpc85xx on x86

Even though qemu have ppce500_mpc8544ds board inside, it relies on kvm to manage MMU stuff. Thus to run it one need PPC host (correct me if I'm wrong). Let's try to emulate BookE MMU in qemu and boot linux.

Plan

- build ppc linux kernel for mpc8544;
- build qemu that can boot this kernel;
- try to boot it and fix all that's broken along the way;

PPC kernel

- need dtc to compile it;
- description of FDT and how kernel expects it: Documentation/powerpc/booting-without-of.txt;
- modify arch/powerpc/boot/wrapper – link_address is too low (0x400000 → 0x800000);
- boot sequence:

```
arch/powerpc/boot/crt0.S (basic platform init)
arch/powerpc/boot/main.c (unzip kernel to PA 0 and jump to it)
arch/powerpc/kernel/head_fsl_booke.S (MMU/vectors setup)
```

- use x/s sprint_buf to see what kernel prints (unless you see serial port output – I don't);

Qemu that can boot -M mpc8544ds -kernel

- need libfdt: take it from scripts/dtc/libfdt, use Makefile:

```
# Makefile.libfdt
#
# This is not a complete Makefile of itself.  Instead, it is designed to
# be easily embeddable into other systems of Makefiles.
#
LIBFDT_INCLUDES = fdt.h libfdt.h
LIBFDT_SRCS = fdt.c fdt_ro.c fdt_wip.c fdt_sw.c fdt_rw.c fdt_strerror.c
LIBFDT_OBJS = $(LIBFDT_SRCS:%.c=%.o)

LIBFDT_SRCS = fdt.c fdt_ro.c fdt_wip.c fdt_sw.c fdt_rw.c fdt_strerror.c
LIBFDT_INCLUDES = fdt.h libfdt.h
LIBFDT_EXTRA = libfdt_internal.h
LIBFDT_LIB = libfdt/libfdt.a
LIBFDT_SHARED_LIB=libfdt.so

LIBFDT_OBJS = $(LIBFDT_SRCS:%.c=%.o)
```

```
CFLAGS += -fPIC -I .

$(LIBFDT_objdir)/$(LIBFDT_LIB): $(addprefix
$(LIBFDT_objdir)/,$(LIBFDT_OBJS))

$(LIBFDT_SHARED_LIB): $(LIBFDT_OBJS)
        $(CC) -shared -o $(LIBFDT_SHARED_LIB) $^
clean:
        rm -f $(LIBFDT_SHARED_LIB) $(LIBFDT_OBJS)
```

- install libfdt.so, fdt.h, libfdt.h, libfdt_env.h
- configure qemu: ../qemu/configure –enable-system –disable-linux-user –enable-fdt –target-list=ppc-softmmu –prefix=`pwd`/root “$@”
- apply some patches :TODO:

Next steps

Ok, we've got to _start, but somehow stuck here.

- what's the initial MMU setup on entry to kernel? see arch/powerpc/kvm/e500_tlb.c, esp. kvmppc_e500_tlb_setup;
- why exception in the first mfmsr? mmu_idx, mem_idx – ??
- implement tlbsx (done), tlbre (done), tlbwe (done), tlbivax (done) for BookE (needed by arch/powerpc/kernel/head_fsl_booke.S)
- how softmmu works? how to substitute existing PPC softmmu by TLB-based one?
- what is hardware TLB entry replacement policy/hash function?
- need to refactor the whole TLB thing for BookE PPC

References

- PowerISA 2.06
- E500 core reference manual (e500corerm.pdf)