```
#!/usr/bin/env python3

# This file is Copyright (c) 2015-2019 Florent Kermarrec <florent@enjoy-digital.fr>
# License: BSD

import argparse

from migen import *

from litex_boards.platforms import de10nano

from litex.soc.integration.soc_sdram import *
from litex.soc.integration.builder import *

from litedram.modules import IS42S16160
from litedram.phy import GENSDRPHY

# CRG ----------------------------------------------------------------------------------------------

class _CRG(Module):
    def __init__(self, platform):
        self.clock_domains.cd_sys    = ClockDomain()
        self.clock_domains.cd_sys_ps = ClockDomain()
        self.clock_domains.cd_por    = ClockDomain(reset_less=True)

        # # #

        # Power on reset
        rst_n = Signal()
        self.sync.por += rst_n.eq(1)
        self.comb += [
            self.cd_por.clk.eq(self.cd_sys.clk),
            self.cd_sys.rst.eq(~rst_n),
            self.cd_sys_ps.rst.eq(~rst_n)
        ]

        # Sys Clk / SDRAM Clk
        clk50 = platform.request("clk50")
        self.comb += self.cd_sys.clk.eq(clk50)
        self.specials += \
            Instance("ALTPLL",
                p_BANDWIDTH_TYPE         = "AUTO",
                p_CLK0_DIVIDE_BY         = 1,
                p_CLK0_DUTY_CYCLE        = 50,
                p_CLK0_MULTIPLY_BY       = 1,
                p_CLK0_PHASE_SHIFT       = "-3000",
                p_COMPENSATE_CLOCK       = "CLK0",
                p_INCLK0_INPUT_FREQUENCY = 20000,
                p_OPERATION_MODE         = "ZERO_DELAY_BUFFER",
                i_INCLK                  = clk50,
                o_CLK                    = self.cd_sys_ps.clk,
                i_ARESET                 = ~rst_n,
                i_CLKENA                 = 0x3f,
                i_EXTCLKENA              = 0xf,
                i_FBIN                   = 1,
                i_PFDENA                 = 1,
                i_PLLENA                 = 1,
            )
        self.comb += platform.request("sdram_clock").eq(self.cd_sys_ps.clk)

# BaseSoC ------------------------------------------------------------------------------------------

class BaseSoC(SoCSDRAM):
    def __init__(self, sys_clk_freq=int(50e6), **kwargs):
        assert sys_clk_freq == int(50e6)
        platform = de10nano.Platform()

        # SoCSDRAM ---------------------------------------------------------------------------------
        SoCSDRAM.__init__(self, platform, clk_freq=sys_clk_freq,
            integrated_rom_size=0x8000,
            **kwargs)
```

```
        # CRG ----------------------------------------------------------------------------------
        self.submodules.crg = _CRG(platform)

        # SDR SDRAM ----------------------------------------------------------------------------
        if not self.integrated_main_ram_size:
            self.submodules.sdrphy = GENSDRPHY(platform.request("sdram"))
            sdram_module = IS42S16160(self.clk_freq, "1:1")
            self.register_sdram(self.sdrphy,
                geom_settings   = sdram_module.geom_settings,
                timing_settings = sdram_module.timing_settings)

# Build --------------------------------------------------------------------------------------

def main():
    parser = argparse.ArgumentParser(description="LiteX SoC on DE10 Nano")
    builder_args(parser)
    soc_sdram_args(parser)
    args = parser.parse_args()

    soc = BaseSoC(**soc_sdram_argdict(args))
    builder = Builder(soc, **builder_argdict(args))
    builder.build()


if __name__ == "__main__":
    main()
```