In [1]:
```
import sys
import datetime
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
from pynq import Overlay

ol = Overlay("block_design.bit")
```

```
---------------------------------------------------------------------------
CacheMetadataError                        Traceback (most recent call last)
File /usr/local/share/pynq-venv/lib/python3.10/site-packages/pynq/pl_server/embe
dded_device.py:252, in BitstreamHandler.get_parser(self, partial)
    251 try:
--> 252     parser = self._get_cache()
    253 except CacheMetadataError:

File /usr/local/share/pynq-venv/lib/python3.10/site-packages/pynq/pl_server/embe
dded_device.py:228, in BitstreamHandler._get_cache(self)
    227 else:
--> 228     raise CacheMetadataError(f"No cached metadata present")

CacheMetadataError: No cached metadata present

During handling of the above exception, another exception occurred:

UnexpectedPortTypeError                   Traceback (most recent call last)
Input In [2], in <cell line: 3>()
      1 from pynq import Overlay
----> 3 ol = Overlay("block_design.bit")

File /usr/local/share/pynq-venv/lib/python3.10/site-packages/pynq/overlay.py:31
9, in Overlay.__init__(self, bitfile_name, dtbo, download, ignore_version, devic
e, gen_cache)
    315 super().__init__(bitfile_name, dtbo, partial=False, device=device)
    317 self._register_drivers()
--> 319 self.device.set_bitfile_name(self.bitfile_name)
    320 self.parser = self.device.parser
    322 self.ip_dict = (
    323     self.gpio_dict
    324 ) = (
    325     self.interrupt_controllers
    326 ) = self.interrupt_pins = self.hierarchy_dict = dict()

File /usr/local/share/pynq-venv/lib/python3.10/site-packages/pynq/pl_server/devi
ce.py:118, in Device.set_bitfile_name(self, bitfile_name)
    116 def set_bitfile_name(self, bitfile_name: str) -> None:
    117     self.bitfile_name = bitfile_name
--> 118     self.parser = self.get_bitfile_metadata(self.bitfile_name)
    119     self.mem_dict = self.parser.mem_dict
    120     self.ip_dict = self.parser.ip_dict

File /usr/local/share/pynq-venv/lib/python3.10/site-packages/pynq/pl_server/embe
dded_device.py:690, in EmbeddedDevice.get_bitfile_metadata(self, bitfile_name, p
artial)
    689 def get_bitfile_metadata(self, bitfile_name:str, partial:bool=False):
--> 690     parser = _get_bitstream_handler(bitfile_name).get_parser(partial=par
tial)
    691     if parser is None:
    692         raise RuntimeError("Unable to find metadata for bitstream")

File /usr/local/share/pynq-venv/lib/python3.10/site-packages/pynq/pl_server/embe
dded_device.py:254, in BitstreamHandler.get_parser(self, partial)
    252     parser = self._get_cache()
    253 except CacheMetadataError:
--> 254     parser = RuntimeMetadataParser(Metadata(input=self._filepath.with_su
ffix(".hwh")))
    255 except:
    256     raise RuntimeError(f"Unable to parse metadata")

File /usr/local/share/pynq-venv/lib/python3.10/site-packages/pynqmetadata/fronte
```

```
nds/metadata.py:39, in Metadata(input)
     37 if os.path.isfile(input):
     38     if str(input).endswith(".hwh"):
---> 39         return HwhFrontend(_hwhfile=input)
     40     elif str(input).endswith(".xsa"):
     41         return XsaFrontend(input=input)

File <string>:25, in __init__(self, name, type, generic_type, _parent, _childre
n, ref, ext, _timestamp, hierarchy_name, ports, parameters, blocks, modules, bus
ses, _hierarchies, _hwhfile, _element_tree, _root, _logical2physical_portmap, _p
hysical2logical_portmap, _logical2physical_extern_pm, _physical2logical_extern_p
m)

File /usr/local/share/pynq-venv/lib/python3.10/site-packages/pynqmetadata/fronte
nds/hwh_frontend.py:219, in HwhFrontend.__post_init__(self)
    207 """
    208 Performs the parsing of the hwh into the metadata model
    209 * checks to see if the hwhfile is an XML string or a
   (...)
    216 * Performs a connectivity pass
    217 """
    218 if self._hwhfile != "":
--> 219     self.parse()

File /usr/local/share/pynq-venv/lib/python3.10/site-packages/pynqmetadata/fronte
nds/hwh_frontend.py:240, in HwhFrontend.parse(self)
    237 self._construct_logical2physical_extern_pm()
    238 self._create_external_ports()
--> 240 self.resolve_addressing()
    241 self.connect_signals()
    243 self.refresh()

File /usr/local/share/pynq-venv/lib/python3.10/site-packages/pynqmetadata/fronte
nds/hwh_frontend.py:603, in HwhFrontend.resolve_addressing(self)
    596 def resolve_addressing(self) -> None:
    597     """
    598     For all the subordinate ports in the design and manager ports
    599     grab all the addressing information
    600     WARNING: This should only be called after all the cores and ports
    601     have been populated.
    602     """
--> 603     self._resolve_subordinate_addressing()
    604     self._populate_subordinate_regmap()
    605     self._resolve_manager_address_maps()

File /usr/local/share/pynq-venv/lib/python3.10/site-packages/pynqmetadata/fronte
nds/hwh_frontend.py:465, in HwhFrontend._resolve_subordinate_addressing(self)
    463     pass
    464 else:
--> 465     raise UnexpectedPortTypeError(
    466         f"Expected {port.ref} to be SubordinatePort when assigning base
address"
    467     )

UnexpectedPortTypeError: Expected block_design:pl_ddr4_0[port] to be Subordinate
Port when assigning base address
```

In [ ]: